

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ  
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ  
імені ІГОРЯ СІКОРСЬКОГО»

*Факультет інформатики та обчислювальної техніки*

(повне найменування інституту, факультету)

*Автоматизованих систем обробки інформації і управління*

(повна назва кафедри)

«До захисту допущено»

**В.о. завідувача кафедри**

\_\_\_\_\_  
(підпис) О.А.Павлов  
(ініціали, прізвище)

“ ” 2019 р.

**Дипломний проект**

**на здобуття ступеня бакалавра**

з напрямку підготовки 6.050103 «Програмна інженерія»

на тему Програмне забезпечення оцінювання результатів наукової

діяльності викладачів

**Виконав: студент IV  
курсу, групи**

ІП-51 Гладушко Валентин Юрійович  
(прізвище, ім'я, по батькові)

\_\_\_\_\_  
(підпис)

**Керівник**

Доц., к.т.н., доц. Фіногенов О.Д.  
посада, науковий ступінь, вчене  
звання, прізвище, ініціали

\_\_\_\_\_  
(підпис)

**Консультант  
з графічної  
документації**

Доц., к.т.н. Ліщук К.І.  
посада, науковий ступінь, вчене  
звання, прізвище, ініціали

\_\_\_\_\_  
(підпис)

**Рецензент:**

Доцент к.т.н., доцент Мелкумян К.Ю.

посада, науковий ступінь, вчене звання, прізвище, ініціали

\_\_\_\_\_  
(підпис)

Засвідчую, що у цьому дипломному проекті  
немає запозичень з праць інших авторів без  
відповідних посилань.

Студент \_\_\_\_\_  
(підпис)

Київ – 2019 року

**Національний технічний університет України  
“Київський політехнічний інститут імені Ігоря Сікорського”**

Факультет (інститут) Інформатики та обчислювальної техніки  
(повна назва)

Кафедра автоматизованих систем обробки інформації і управління  
(повна назва)

Рівень вищої освіти – перший (бакалаврський)

Напрямок підготовки (програма професійного спрямування) – 6.050103  
«Програмна інженерія» (Програмне забезпечення систем)

**ЗАТВЕРДЖУЮ**

**В.о. завідувача кафедри**

О.А. Павлов  
(підпис) (ініціали, прізвище)

“ ” 2019 р.

**ЗАВДАННЯ  
НА ДИПЛОМНИЙ ПРОЕКТ СТУДЕНТУ**

Гладишук Валентину Юрійовичу  
(прізвище, ім'я, по батькові)

**1. Тема проекту** «Програмне забезпечення оцінювання результатів наукової діяльності викладачів»

керівник проекту Олексій Дмитрович Фіногенов, к.т.н., доцент  
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом по університету від “23” квітня 2019 р. №1181-с

**2. Термін подання студентом проекту** «03» червня 2019 року

**3. Вихідні дані до проекту**

*Технічне завдання*

**4. Зміст пояснювальної записки**

*1) Аналіз вимог до програмного забезпечення: основні визначення та терміни, опис предметного середовища, огляд існуючих технічних рішень та відомих програмних продуктів, розробка функціональних та нефункціональних вимог*

*2) Моделювання та конструювання програмного забезпечення: моделювання та аналіз програмного забезпечення, засоби розробки, технічні рішення, архітектура*

*3) Аналіз якості та тестування програмного забезпечення*

*4) Впровадження та супровід програмного забезпечення. Керівництво користувача.*

## 5. Перелік графічного матеріалу

1) *Схема бази даних*

2) *Схема структурна класів програмного забезпечення*

3) *Креслення вигляду екранних форм*

## 6. Консультанти розділів проекту

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв

7. Дата видачі завдання «12» березня 2018 року

## КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів виконання дипломного Проекту	Термін виконання етапів проекту	Примітка
1.	<i>Вивчення рекомендованої літератури</i>	<i>20.04.2019</i>	
2.	<i>Аналіз існуючих методів розв'язання задачі</i>	<i>20.04.2019</i>	
3.	<i>Постановка та формалізація задачі</i>	<i>25.08.2019</i>	
4.	<i>Аналіз вимог до програмного забезпечення</i>	<i>25.04.2019</i>	
5.	<i>Алгоритмізація задачі</i>	<i>28.04.2019</i>	
6.	<i>Моделювання програмного забезпечення</i>	<i>28.04.2019</i>	
7.	<i>Обґрунтування використовуваних технічних засобів</i>	<i>28.04.2019</i>	
8.	<i>Розробка архітектури програмного забезпечення</i>	<i>28.04.2019</i>	
9.	<i>Розробка програмного забезпечення</i>	<i>15.05.2019</i>	
10.	<i>Налагодження програми</i>	<i>19.05.2019</i>	
11.	<i>Виконання графічних документів</i>	<i>25.05.2019</i>	
12.	<i>Оформлення пояснювальної записки</i>	<i>25.05.2019</i>	
13.	<i>Подання ДП на попередній захист</i>	<i>28.05.2019</i>	
14.	<i>Подання ДП рецензенту</i>	<i>03.06.2019</i>	
15.	<i>Подання ДП на основний захист</i>	<i>08.06.2019</i>	

Студент \_\_\_\_\_ Гладишко В.Ю.  
(підпис)

Керівник проекту \_\_\_\_\_ Фіногенов О.Д.  
(підпис)

[illegible]

## АНОТАЦІЯ

**Структура та обсяг роботи.** Пояснювальна записка дипломного проекту складається з чотирьох розділів, містить 61 сторінку, 9 рисунків, 41 таблицю, 4 додатки, 15 джерел.

Дипломний проект присвячений розробці програмного забезпечення для оцінювання результатів наукової діяльності викладачів в вищих навчальних закладах.

Мета дипломного проекту: створити настільний додаток для отримання оцінок результатів наукової діяльності викладачів та різноманітної статистики по оцінках наукової діяльності.

У першому розділі були визначені функціональні нефункціональні вимоги до програмного забезпечення, також були розглянуті основні конкурентні IT-проекти.

У другому розділі були описані діаграми бізнес процесів, була розроблена архітектура програмного забезпечення.

У третьому розділі були розглянуті основні процеси тестування.

У четвертому розділі було описано процес розгортання програмного забезпечення.

У додатках наведено: інструкцію користувача, діаграма класів.

**КЛЮЧОВІ СЛОВА:** ІНДЕКС ГІРША, НАСТІЛЬНИЙ ДОДАТОК, ОЦІНЮВАННЯ РЕЗУЛЬТАТІВ НАУКОВОЇ ДІЯЛЬНОСТІ.

## ABSTRACT

**Structure and scope of work.** The explanatory note of the diploma project consists of four sections containing 61 pages, 9 figures, 41 tables, 4 additions, 15 sources.

The diploma project devotes to the development of software for assessing the results of academic activities of teachers in educational institutions.

The purpose of the diplomatic project: to create the easiest application for evaluating the results of scientific activities and various statistics on scientific activities.

In the first section, functional non-functional software requirements were identified, and major competitive IT projects were considered.

In the second section, the business processes charts were described, and the software architecture was developed.

In the third section, the basic testing processes were considered.

The fourth section describes the process of deploying software.

**KEY WORDS:** HIRSCH INDEX, DESKTOP APPLICATION, EVALUATION OF RESULTS OF SCIENTIFIC ACTIVITIES.

					КПІ.ІП-5104.045490.00.91	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		6

# **Пояснювальна записка до дипломного проекту**

на тему: Програмне забезпечення оцінювання результатів наукової  
діяльності викладачів

Київ – 2019 року

## ЗМІСТ

<b>ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ.....</b>	<b>10</b>
<b>ВСТУП.....</b>	<b>11</b>
<b>1 АНАЛІЗ ВИМОГ ДО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ.....</b>	<b>12</b>
1.1 ЗАГАЛЬНІ ПОЛОЖЕННЯ.....	12
1.2 ЗМІСТОВНИЙ ОПИС І АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ.....	12
1.3 АНАЛІЗ УСПІШНИХ ІТ-ПРОЕКТІВ.....	13
1.3.1 Google Scholar.....	13
1.3.2 Scopus.....	14
1.3.3 Web of Science .....	14
1.3.4 Orcid.....	15
1.4 АНАЛІЗ ВИМОГ ДО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ.....	15
1.4.1 Розроблення функціональних вимог.....	15
1.4.2 Розроблення нефункціональних вимог.....	26
1.4.3 Постановка комплексу завдань модулю .....	27
1.5 Висновки по розділу .....	27
<b>2 МОДЕЛЮВАННЯ ТА КОНСТРУЮВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ.....</b>	<b>28</b>
2.1 МОДЕЛЮВАННЯ ТА АНАЛІЗ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ.....	28
2.2 АРХІТЕКТУРА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ .....	33
2.3 ОПИС СТРУКТУРИ БАЗИ ДАНИХ .....	34
2.4 КОНСТРУЮВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ.....	40
2.5 АНАЛІЗ БЕЗПЕКИ ДАНИХ.....	55
2.6 Висновки по розділу .....	55
<b>3 АНАЛІЗ ЯКОСТІ ТА ТЕСТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ.....</b>	<b>56</b>
3.1 АНАЛІЗ ЯКОСТІ ПЗ.....	56
3.2 ОПИС ПРОЦЕСІВ ТЕСТУВАННЯ.....	57
3.3 ОПИС КОНТРОЛЬНОГО ПРИКЛАДУ .....	57



3.4	Висновок до розділу .....	58
-----	---------------------------	----

## 4 ВПРОВАДЖЕННЯ ТА СУПРОВІД ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ..... 59

4.1	Розгортання програмного забезпечення.....	59
-----	---	----

4.1.1	Встановлення СКБД MS SQL Server.....	60
-------	--------------------------------------	----

4.1.2	Створення бази даних.....	60
-------	---------------------------	----

4.1.3	Встановлення настільного додатку KpiResearchersEvaluation .....	60
-------	---	----

4.2	Робота з програмним забезпеченням.....	60
-----	--	----

## ВИСНОВКИ ..... 61

## ПЕРЕЛІК ПОСИЛАНЬ ..... 62

					КПІ.ІП-5104.045490.01.81	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		9

## ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

API (Application Programming Interface) – це опис способів (набір класів, функцій, інтерфейсів) взаємодії з системою з інших програм.

БД (База даних) – це представлення даних в об'єктивній формі, яка описує взаємозв'язки між їх елементами і характеристику цих даних.

BRMN – це система умовних позначень, за допомогою нотації UML, для моделювання бізнес процесів.

XAML (eXtensible Application Markup Language) – це декларативна мова розмітки, яка використовується в .NET Framework.

СКБД (Система керування базами даних) – це набір з бази даних та програм для взаємодії з цими даними.

WPF (Windows Presentation Foundation) – це система для створення клієнтських додатків Windows з візуальними способами взаємодії з користувачем.

MS SQL Server – це СКБД, розроблена корпорацією Microsoft, мова, яка використовується для запитів - Transact-SQL.

## ВСТУП

Оцінювання результатів наукової діяльності викладачів є однією з основних складових навчального процесу. На основі даних оцінок створюються різноманітні рейтинги вищих навчальних закладів, і завдяки цим рейтингам можна оцінити науковий вклад та престижність вищих навчальних закладів.

Для оцінювання результатів наукової діяльності викладачів існують різноманітні загальнодоступні наукометричні бази даних в Інтернеті, наприклад Google Scholar [1], Scopus [2], Web of Science (Publons) [3]. В даних базах використовується метрика індекс Гірша [4], в Google Scholar також використовується i10 індекс. Ці метрики засновані на кількості публікацій і кількості цитувань цих публікацій. Недоліком цих баз даних є те що вони оцінюють кожного наукового діяча окремо, тож за допомогою них неможна оцінити наукову діяльність цілої кафедри, факультету або навчального закладу. Також в цих базах даних відсутня інформація про кафедру та факультет до якого належить викладач.

Для зручного оцінювання результатів наукової діяльності викладачів КПІ та складання рейтингів ефективності потрібен інструмент котрий дозволить дізнатись ефективність обраної кафедри чи факультету, а також всіх викладачів, які до неї належать.

Саму тому потрібно створити настільний додаток, котрий буде зчитувати дані по викладачах КПІ, їх публікаціям та кількості цитувань з веб-ресурсів Google Scholar, Scopus, Web of Science, а потім буде представляти інформацію в зручному форматі.

Метою розробки є:

- автоматизація збору даних із ресурсів Google Scholar, Scopus та Web of Science;
- визначення результатів наукової діяльності викладачів, кафедр та факультетів, що представлені в бібліографічних базах даних.

					КПІ.ІП-5104.045490.01.81	Арк.
						11
Змн.	Арк.	№ докум.	Підпис	Дата		

# 1 АНАЛІЗ ВИМОГ ДО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

## 1.1 Загальні положення

Оцінювання результатів наукової діяльності є дуже важливим для науки в наш час. Без оцінювання не можна було б дізнатись науковий потенціал наукового діяча, кафедри, університету, наукового видання чи країни.

Для того щоб визначати ефективність наукових діячів на разі створено декілька великим бібліографічних баз даних, та створено безліч різноманітних метрик. Тому потрібно визначити які з цих інструментів використовувати.

## 1.2 Змістовний опис і аналіз предметної області

Для вимірювання ефективності наукової діяльності викладачів використовуються різні наукометричні показники.

Найбільш розповсюдженими показниками є загальна кількість цитувань, загальна кількість статей, індекс Гірша та i10 індекс.

Загальна кількість статей не враховує якість наукових публікацій, в той час як на загальну кількість цитувань може непропорційно вплинути участь однієї публікації з істотним впливом (наприклад, методологічні статті, що пропонують успішні нові технології або методи, які можуть генерувати велику кількість цитат), або багато публікацій з малою кількістю цитат.

Індекс Гірша спрямований на те, щоб усунути основні недоліки інших бібліометричних показників, таких як загальна кількість статей або загальна кількість посилань. Він враховує одночасно і кількість, і якість наукових робіт.

Індекс Гірша заснований на наборі найбільш цитованих праць наукового діяча і кількості цитат, які вони отримали в інших публікаціях. Індекс також може бути застосований до продуктивності і впливу наукового журналу, а також групи вчених, таких як факультет, університет або країна.

Індекс визначається як максимальне значення  $h$ , таке що даний автор опублікував  $h$  статей, кожна з яких була процитована не менше  $h$  раз.

					КП.ІП-5104.045490.01.81	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		12

Основними недоліком індексу Гірша є те що його не можна використовувати для порівняння результатів наукової діяльності викладачів з різних наукових областей (тобто типове значення індексу для Математики та Біології буде відрізнятись).

Також використовується i10 індекс.

Він використовується в Google Scholar для ефективності наукової діяльності наукових діячів.

Індекс визначається як кількість публікацій, що мають 10 або більше бібліографічних посилань.

Основним недоліком цього індексу є те, що він не придатний для оцінки наукової діяльності молодих науковців, що мають невелике число цитувань, і при цьому, жодна з публікацій ще не має принаймні 10 цитувань. Також він не придатний для порівняння результатів наукової діяльності викладачів з різних наукових областей.

### 1.3 Аналіз успішних IT-проектів

Основними конкурентами даного програмного забезпечення є різні веб-ресурси наукометричні бази даних, такі як Google Scholar, Scopus, Web of Science та Orcid [5]. В цих бібліометричних базах даних зазвичай міститься стисла інформація про наукових діячів та їх публікації.

#### 1.3.1 Google Scholar

Це безкоштовна пошукова система яка індексує повні тексти наукових публікацій наукових діячів. Індксація наукових статей в Google Scholar виконується по даних з більшості рецензованих онлайн журналів, які належать найбільшим науковим видавництвам Європи та Америки.

В Google Scholar можна виконувати пошук по статтям доступним тільки в бібліотеках або онлайн. Також в Google Scholar можна отримати інформацію

про викладача, стислу інформацію про його публікації а також його індекс Гірша та i10 індекс.

В Google Scholar всі дані в відкритому доступі, тобто непотрібна платна підписка, але відсутній API для отримання даних.

### 1.3.2 Scopus

Це бібліографічна база даних для оцінювання наукової ваги публікацій заснованої на кількості посилань. В Scopus відбувається індексація серійних книжкових видань, матеріалів конференцій, наукових журналів, а також професійних журналів.

Доступ до даних відбувається через веб-інтерфейс на умовах платної підписки. Також є обмежений доступ до даних безкоштовно за допомогою Scopus Author Preview.

В Scopus надається така інформація про наукових діячів: повне ім'я діяча, перелік його наукових робіт та кількість цитувань для кожної із них, а також індекс Гірша даного діяча.

Кожен науковий діяч має унікальний AuthorID в цій базі даних, за допомогою даного ідентифікатора можна швидко знайти заданого автора.

Scopus також має закритий, доступний за підпискою, API для отримання даних.

### 1.3.3 Web of Science

Це наукометрична база даних, яка індексує публікації за допомогою шести різних онлайн-баз даних.

Цей програмний продукт декілька разів змінював свої назву, раніше він називався Web of Knowledge, зараз відбувається перехід на нову адресу Publons.

Web of Science містить всі дані в відкритому доступі, але API має доступ за підпискою.

Дані про публікації збираються з таких баз даних:

					КП.ІП-5104.045490.01.81	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		14

- розширений індекс наукового цитування;
- індекс цитування соціальних наук;
- індекс цитування по мистецтву і гуманітарним наукам;
- індекс цитування нових джерел;
- індекс цитування книг;
- індекс цитування матеріалів конференцій.

Всі наукові діячі мають унікальні ідентифікатори ResearcherID та PublonsID, також даний веб-ресурс містить інформацію про індекс Гірша даного діяча.

#### 1.3.4 Orcid

Цей веб-ресурс був створений для того щоб об'єднати декілька записів в різних бібліографічних баз в один профіль.

Кожен науковий діяч може отримати ідентифікатор Orcid безкоштовно, діяч може в разі необхідності змінити дані про ім'я та додати дані про ідентифікатори з Web of Science та Scopus, та посилання на профілі в цих базах.

В Orcid є відкритий API для доступу до даних публічних аккаунтів.

На жаль в даному веб-ресурсі не зберігається інформація про індекс Гірша.

#### 1.4 Аналіз вимог до програмного забезпечення

Аналізуючи наявний функціонал основних конкурентів, було сформовано ряд функціональних та не функціональних вимог, які необхідні для створення програмного забезпечення.

##### 1.4.1 Розроблення функціональних вимог

Система взаємодіє з такими учасниками: Працівник КПІ.

Варіанти використання наведені в таблицях:

					КПІ.ІП-5104.045490.01.81	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		15

Таблиця 1.1 – Варіант використання UC001

Назва	Додавання факультету.
Опис	Користувач додає інформацію про факультет.
Учасники	Працівник КП.
Передумови	
Постумови	Успішне створення запису в БД.
Основний сценарій	<ol style="list-style-type: none"> <li>1) Користувач відриває настільний додаток;</li> <li>2) Користувач обирає вкладку «Факультети»;</li> <li>3) Користувач нажимає на кнопку «Створити факультет»;</li> <li>4) Система відкриває вікно для створення факультету;</li> <li>5) Користувач заповнює дані про факультет.</li> <li>6) Користувач нажимає на кнопку «Створити»;</li> <li>7) Система зберігає факультет в БД.</li> </ol>
Розширення сценаріїв	<ol style="list-style-type: none"> <li>6.1) Дані не проходять валідацію;</li> <li>6.1.а) Система вказує користувачу на неправильно заповнені поля.</li> </ol>

Таблиця 1.2 – Варіант використання UC002

Назва	Редагування факультету.
Опис	Користувач редагує інформацію про факультет.
Учасники	Працівник КП.



## Продовження таблиці 1.2

Передумови	Існування запису в БД.
Постумови	Успішне редагування запису в БД.
Основний сценарій	<ol style="list-style-type: none"> <li>1) Користувач відриває настільний додаток;</li> <li>2) Користувач обирає вкладку «Факультети»;</li> <li>3) Користувач обирає факультет;</li> <li>4) Користувач нажимає на кнопку «Редагувати факультет»;</li> <li>5) Система відкриває вікно для редагування факультету;</li> <li>6) Користувач редагує дані про факультет.</li> <li>7) Користувач нажимає на кнопку «Редагувати»;</li> <li>8) Система зберігає зміни в БД.</li> </ol>
Розширення сценаріїв	<ol style="list-style-type: none"> <li>7.1) Дані не проходять валідацію;</li> <li>7.1.а) Система вказує користувачу на неправильно заповнені поля.</li> </ol>

## Таблиця 1.3 – Варіант використання UC003

Назва	Перегляд інформації про факультет.
Опис	Користувач переглядає інформацію про факультет.
Учасники	Працівник КПП.
Передумови	Існування запису в БД.
Постумови	

## Продовження таблиці 1.3

Основний сценарій	1) Користувач відриває настільний додаток; 2) Користувач обирає вкладку «Факультети»; 3) Користувач обирає факультет; 4) Система відображає інформацію про факультет.
Розширення сценаріїв	

## Таблиця 1.4 – Варіант використання UC004

Назва	Додавання кафедри.
Опис	Користувач додає інформацію про кафедру.
Учасники	Працівник КП.
Передумови	
Постумови	Успішне створення запису в БД.
Основний сценарій	1) Користувач відриває настільний додаток; 2) Користувач обирає вкладку «Кафедри»; 3) Користувач нажимає на кнопку «Створити кафедру»; 4) Система відкриває вікно для створення кафедри; 5) Користувач заповнює дані про кафедру. 6) Користувач нажимає на кнопку «Створити»; 7) Система зберігає кафедру в БД.
Розширення сценаріїв	6.1) Дані не проходять валідацію; 6.1.а) Система вказує користувачу на неправильно заповнені поля.

Таблиця 1.5 – Варіант використання UC005

Назва	Редагування кафедри.
Опис	Користувач редагує інформацію про кафедру.
Учасники	Працівник КПІ.
Передумови	Існування запису в БД.
Постумови	Успішне редагування запису в БД.
Основний сценарій	<ol style="list-style-type: none"> <li>1) Користувач відриває настільний додаток;</li> <li>2) Користувач обирає вкладку «Кафедри»;</li> <li>3) Користувач обирає кафедру;</li> <li>4) Користувач нажимає на кнопку «Редагувати кафедру»;</li> <li>5) Система відкриває вікно для редагування кафедри;</li> <li>6) Користувач редагує дані про кафедру.</li> <li>7) Користувач нажимає на кнопку «Редагувати»;</li> <li>8) Система зберігає зміни в БД.</li> </ol>
Розширення сценаріїв	<ol style="list-style-type: none"> <li>7.1) Дані не проходять валідацію;</li> <li>7.1.а) Система вказує користувачу на неправильно заповнені поля.</li> </ol>

Таблиця 1.6 – Варіант використання UC006

Назва	Перегляд інформації про кафедру.
Опис	Користувач переглядає інформацію про кафедру.
Учасники	Працівник КПІ.

Продовження таблиці 1.6

Передумови	Існування запису в БД.
Постумови	
Основний сценарій	1) Користувач відриває настільний додаток; 2) Користувач обирає вкладку «Кафедри»; 3) Користувач обирає кафедру; 4) Система відображає інформацію про кафедру.
Розширення сценаріїв	

Таблиця 1.7 – Варіант використання UC007

Назва	Додавання викладача.
Опис	Користувач додає інформацію про викладача.
Учасники	Працівник КП.
Передумови	
Постумови	Успішне створення запису в БД.
Основний сценарій	1) Користувач відриває настільний додаток; 2) Користувач обирає вкладку «Викладачі»; 3) Користувач нажимає на кнопку «Створити викладача»; 4) Система відкриває вікно для створення викладача; 5) Користувач заповнює дані про викладача. 6) Користувач нажимає на кнопку «Створити»; 7) Система зберігає викладача в БД.

## Продовження таблиці 1.7

Розширення сценаріїв	6.1) Дані не проходять валідацію; 6.1.а) Система вказує користувачу на неправильно заповнені поля.
----------------------	---

Таблиця 1.8 – Варіант використання UC008

Назва	Редагування викладача.
Опис	Користувач редагує інформацію про викладача.
Учасники	Працівник КП.
Передумови	Існування запису в БД.
Постумови	Успішне редагування запису в БД.
Основний сценарій	1) Користувач відриває настільний додаток; 2) Користувач обирає вкладку «Викладачі»; 3) Користувач обирає викладача; 4) Користувач нажимає на кнопку «Редагувати викладача»; 5) Система відкриває вікно для редагування викладача; 6) Користувач редагує дані про викладача. 7) Користувач нажимає на кнопку «Редагувати»; 8) Система зберігає зміни в БД.
Розширення сценаріїв	7.1) Дані не проходять валідацію; 7.1.а) Система вказує користувачу на неправильно заповнені поля.

Таблиця 1.9 – Варіант використання UC009

Назва	Перегляд інформації про викладача.
Опис	Користувач переглядає інформацію про викладача.
Учасники	Працівник КПП.
Передумови	Існування запису в БД.
Постумови	
Основний сценарій	<ol style="list-style-type: none"> <li>1) Користувач відриває настільний додаток;</li> <li>2) Користувач обирає вкладку «Викладачі»;</li> <li>3) Користувач обирає викладача;</li> <li>4) Система відображає інформацію про викладача.</li> </ol>
Розширення сценаріїв	

Таблиця 1.10 – Варіант використання UC010

Назва	Збереження даних про викладачів.
Опис	Збереження даних про викладачів з веб-ресурсів.
Учасники	Працівник КПП.
Передумови	Існування файлу.
Постумови	Збереження даних в БД.
Основний сценарій	<ol style="list-style-type: none"> <li>1) Користувач відриває настільний додаток;</li> <li>2) Користувач обирає вкладку «Перевірка даних»;</li> <li>3) Користувач нажимає на кнопку «Завантажити дані для збереження»;</li> </ol>

## Продовження таблиці 1.10

Основний сценарій	<p>4) Користувач вибирає файл;</p> <p>5) Система завантажує дані про викладачів, які є в файлі, з веб-ресурсів;</p> <p>6) Система зберігає завантажені дані в базу даних; Система відображає інформацію про збережених викладачів.</p>
Розширення сценаріїв	<p>4.1) Дані в файлі в неправильному форматі;</p> <p>4.1.a) Система повідомляє користувача про бажаний формат.</p>

Таблиця 1.11 – Варіант використання UC011

Назва	Перевірка даних про викладачів.
Опис	Перевірка даних про викладачів з файлу.
Учасники	Працівник КП.
Передумови	Існування файлу, існування даних в БД.
Постумови	
Основний сценарій	<p>1) Користувач відриває настільний додаток;</p> <p>2) Користувач обирає вкладку «Перевірка даних»;</p> <p>3) Користувач нажимає на кнопку «Завантажити дані для перевірки»;</p> <p>4) Користувач вибирає файл;</p> <p>5) Система звіряє дані з файлу та з бази даних;</p> <p>6) Система відображає результати перевірки.</p>

## Продовження таблиці 1.11

Розширення сценаріїв	7.1) Дані не проходять валідацію; 7.1.а) Система вказує користувачу на неправильно заповнені поля.
----------------------	---

## Функціональні вимоги:

Таблиця 1.12 – Опис функціональної вимоги REQ001

Номер	REQ001
Назва	Управління даними про факультети
Опис	Система дозволяє працівнику КП керувати факультетами, а саме видаляти, редагувати та переглядати дані про факультети.

Таблиця 1.13 – Опис функціональної вимоги REQ002

Номер	REQ002
Назва	Управління даними про кафедри
Опис	Система дозволяє працівнику КП керувати кафедрами, а саме видаляти, редагувати та переглядати дані про кафедри.

Таблиця 1.14 – Опис функціональної вимоги REQ003

Номер	REQ003
Назва	Управління даними про викладачів
Опис	Система дозволяє працівнику КП керувати викладачами, а саме видаляти, редагувати та переглядати дані про викладачів.



Таблиця 1.15 – Опис функціональної вимоги REQ004

Номер	REQ004
Назва	Перегляд результатів наукової діяльності викладача
Опис	Система дозволяє працівнику КПІ переглядати індекс Гірша та i10 індекс викладача.

Таблиця 1.16 – Опис функціональної вимоги REQ005

Номер	REQ005
Назва	Перегляд результатів наукової діяльності кафедри
Опис	Система дозволяє працівнику КПІ переглядати індекс Гірша та i10 індекс кафедри.

Таблиця 1.17 – Опис функціональної вимоги REQ006

Номер	REQ006
Назва	Перегляд результатів наукової діяльності факультету
Опис	Система дозволяє працівнику КПІ переглядати індекс Гірша та i10 індекс факультету.

Таблиця 1.18 – Опис функціональної вимоги REQ007

Номер	REQ007
Назва	Завантаження даних про викладачів з веб-ресурсів.
Опис	Система дозволяє працівнику КПІ зберегти дані про викладачів, які є в файлі, з веб-ресурсів.

Таблиця 1.19 – Опис функціональної вимоги REQ007

Номер	REQ008
Назва	Перевірка даних про викладачів з файлу.
Опис	Система дозволяє працівнику КП перевірити дані про викладачів з файлу.

Результуюча матриця трасування зображена на рисунку 1.1

	UC001	UC002	UC003	UC004	UC005	UC006	UC007	UC008	UC009	UC010	UC011
REQ001											
REQ002											
REQ003											
REQ004											
REQ005											
REQ006											
REQ007											
REQ008											

Рисунок 1.1 – Матриця трасування

#### 1.4.2 Розроблення нефункціональних вимог

Програмне забезпечення повинне відповідати наступним нефункціональним вимогам:

- локалізація інтерфейсу – українська;
- програмне забезпечення повинно працювати під управлінням операційних систем сімейства WIN32 (Windows XP, Windows 7, Windows 8, Windows 10).

### 1.4.3 Постановка комплексу завдань модулю

Розроблюване програмне забезпечення призначене для вирішення задачі оцінки результатів наукової діяльності викладачів, кафедр та факультетів.

Мета створення даної роботи – створення настільного додатку для автоматизації збору даних із ресурсів Google Scholar, Scopus та Web of Science та визначення результатів наукової діяльності.

Цілями розробки є:

- автоматизація збору даних із різних ресурсів;
- спрощення оцінювання результатів наукової діяльності кафедр та факультетів.

### 1.5 Висновки по розділу

В першому розділі розглянуті основні підходи до вимірювання ефективності наукової діяльності.

Оскільки наукометричні показники загальна кількість статей, загальна кількість цитувань погано відображають кількість та якість публікацій, було вирішено використовувати індекс Гірша та i10 індекс. Були розглянуті основні конкуренти, їх переваги та недоліки, на їх основі були розроблені основні функціональні та не функціональні вимоги до настільного додатку. На основі вимог була виконана постановка комплексного завдання модулю.

## 2 МОДЕЛЮВАННЯ ТА КОНСТРУЮВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

### 2.1 Моделювання та аналіз програмного забезпечення

Для того, щоб створити зручне програмне забезпечення, спочатку потрібно виконати його детальне моделювання та описати детально архітектуру програмного забезпечення.

Для опису бізнес процесів було використані BPMN діаграми.

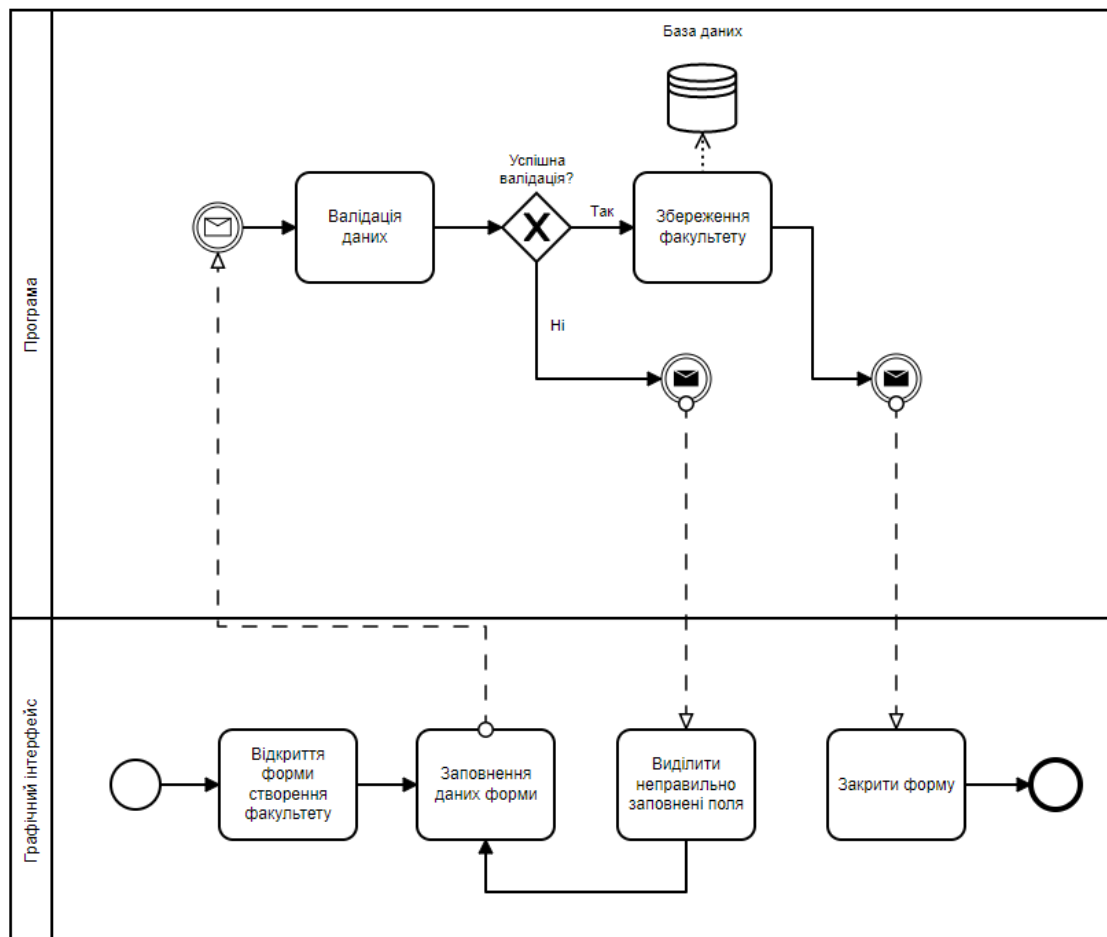


Рисунок 2.1 – Схема процесу створення факультету

Опис процесу створення факультету:

- користувач відкриває форму створення факультету;
- користувач вводить ім'я факультету в відповідне поле;

- користувач натискає кнопку «створити»;
- система валідує введені дані;
- якщо валідація даних не успішна, система виділяє неправильно введені поля;
- інакше система зберігає дані про факультет в БД та закриває форму.

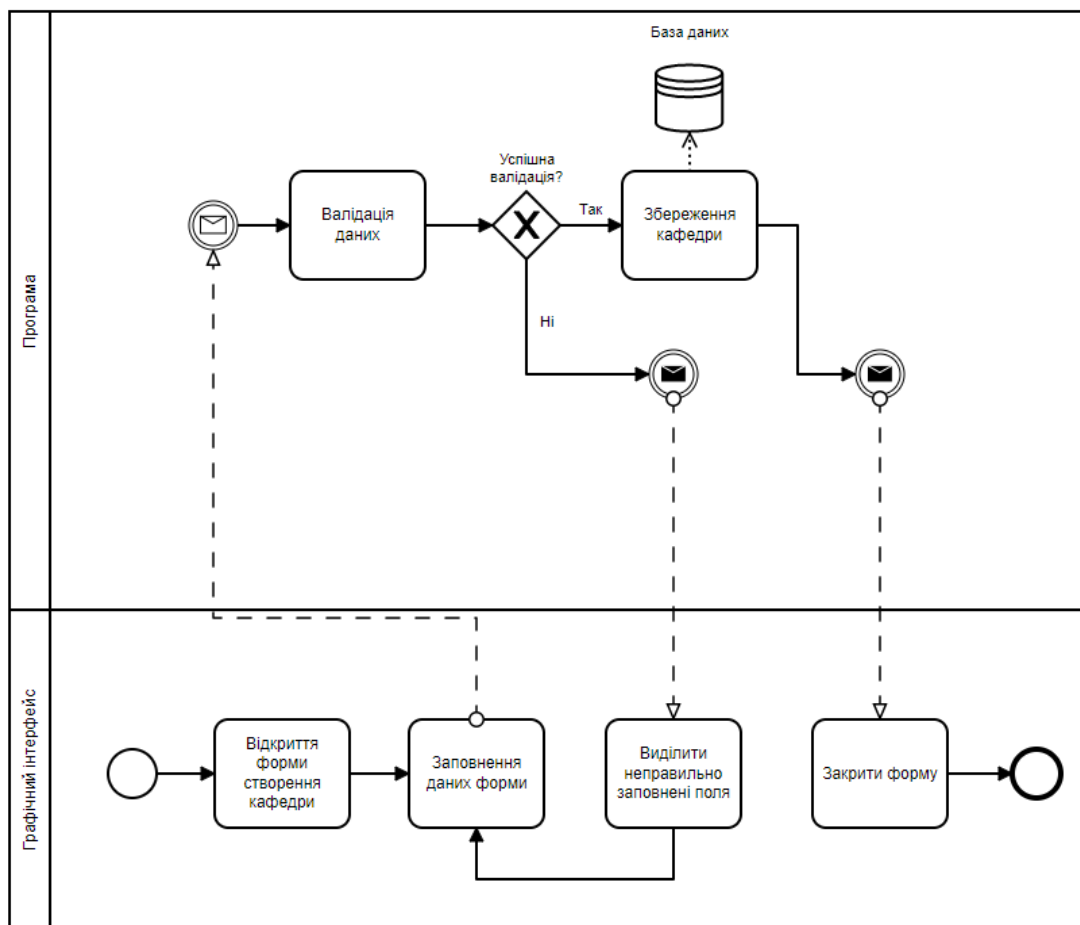


Рисунок 2.2 – Схема процесу створення кафедри

Опис процесу створення кафедри:

- користувач відкриває форму створення кафедри;
- користувач вводить ім'я кафедри в відповідне поле, та вибирає факультет до якого належить кафедра;
- користувач натискає кнопку «Створити»;
- система валідує введені дані;

- якщо валідація даних не успішна, система виділяє неправильно введені поля;
- інакше система зберігає дані про кафедру в БД та закриває форму.

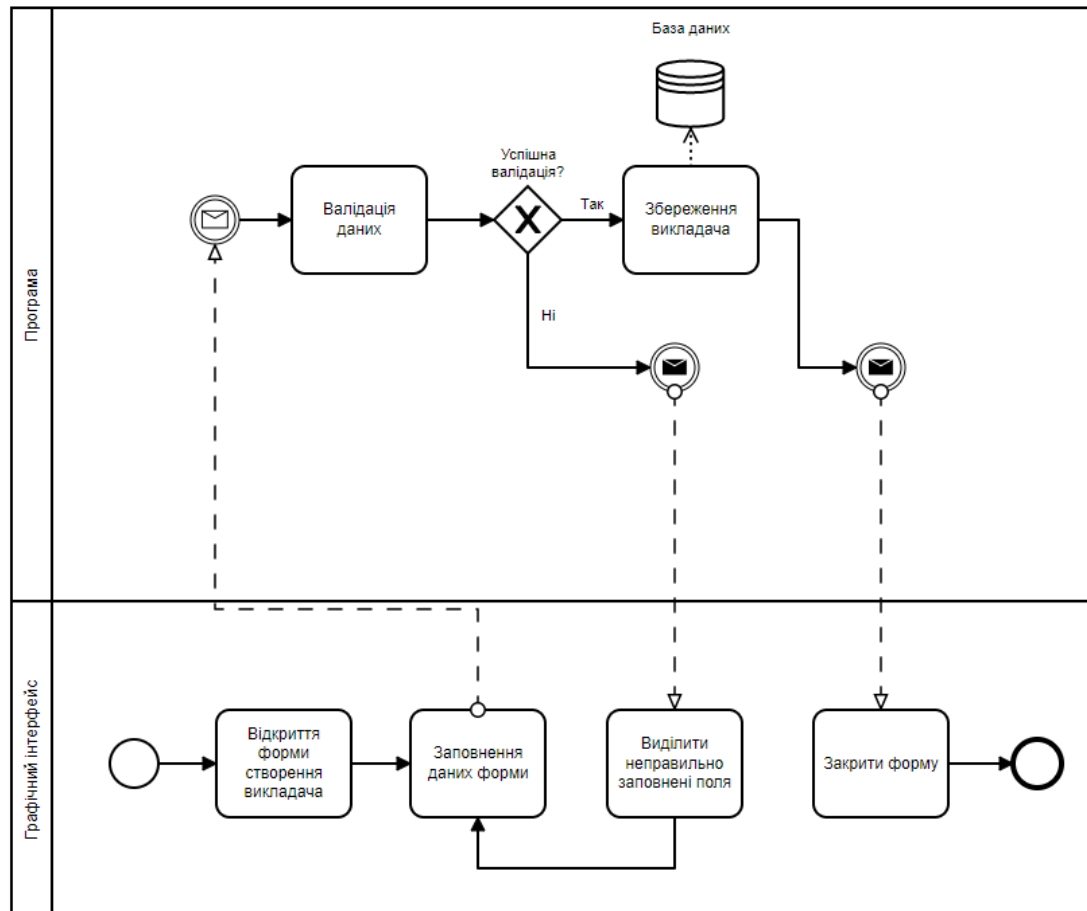


Рисунок 2.3 – Схема процесу створення викладача

Опис процесу створення викладача:

- користувач відкриває форму створення викладача;
- користувач вводить ПІБ викладача в відповідні поля, та вибирає кафедру до якої належить викладач;
- користувач натискає кнопку «Створити»;
- система валідує введені дані;
- якщо валідація даних не успішна, система виділяє неправильно введені поля;

- інакше система зберігає дані про викладача в БД та закриває форму.

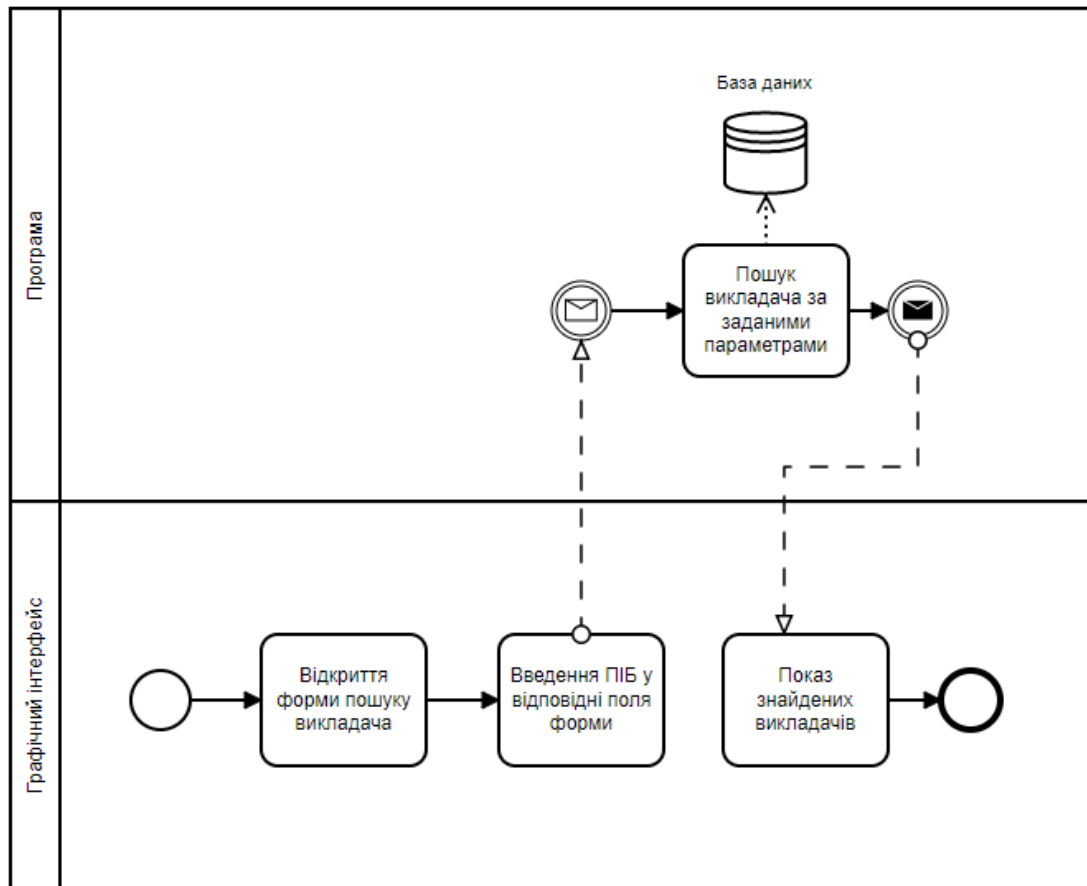


Рисунок 2.4 – Схема процесу пошуку викладача

Опис процесу пошуку викладача:

- користувач відкриває форму пошуку викладача;
- користувач вводить ПІБ викладача в поля пошуку;
- користувач натискає кнопку «Пошук»;
- система шукає викладачів в БД за заданими параметрами;
- система відображає результати пошуку.

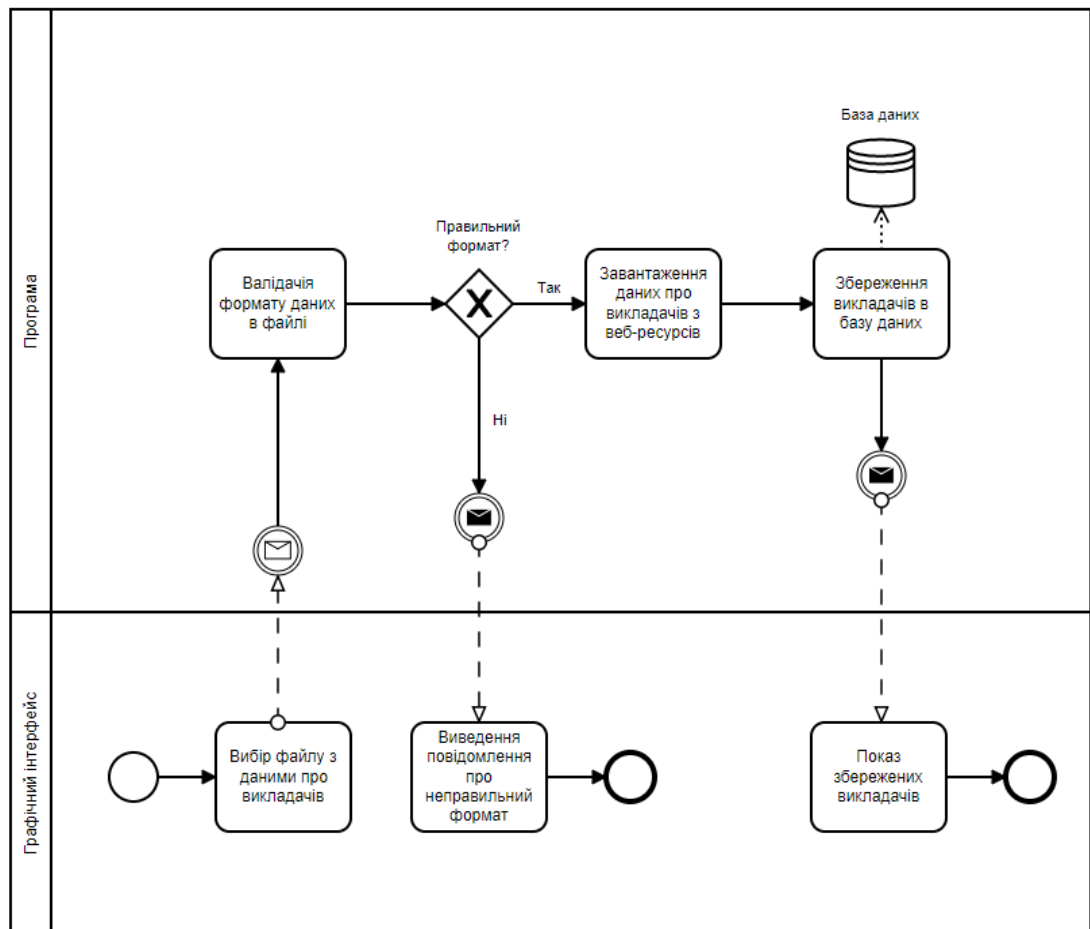


Рисунок 2.5 – Схема процесу збереження даних викладачів з веб-ресурсів

Опис процесу збереження даних викладачів з веб-ресурсів:

- користувач вибирає файл з даними про викладачів;
- система валідує формат даних в файлі;
- якщо валідація даних не успішна, система виводить повідомлення про неправильний формат;
- інакше система завантажує дані про викладачів з веб-ресурсів;
- далі система зберігає дані про викладачів в БД;
- далі система відображає збережених викладачів.



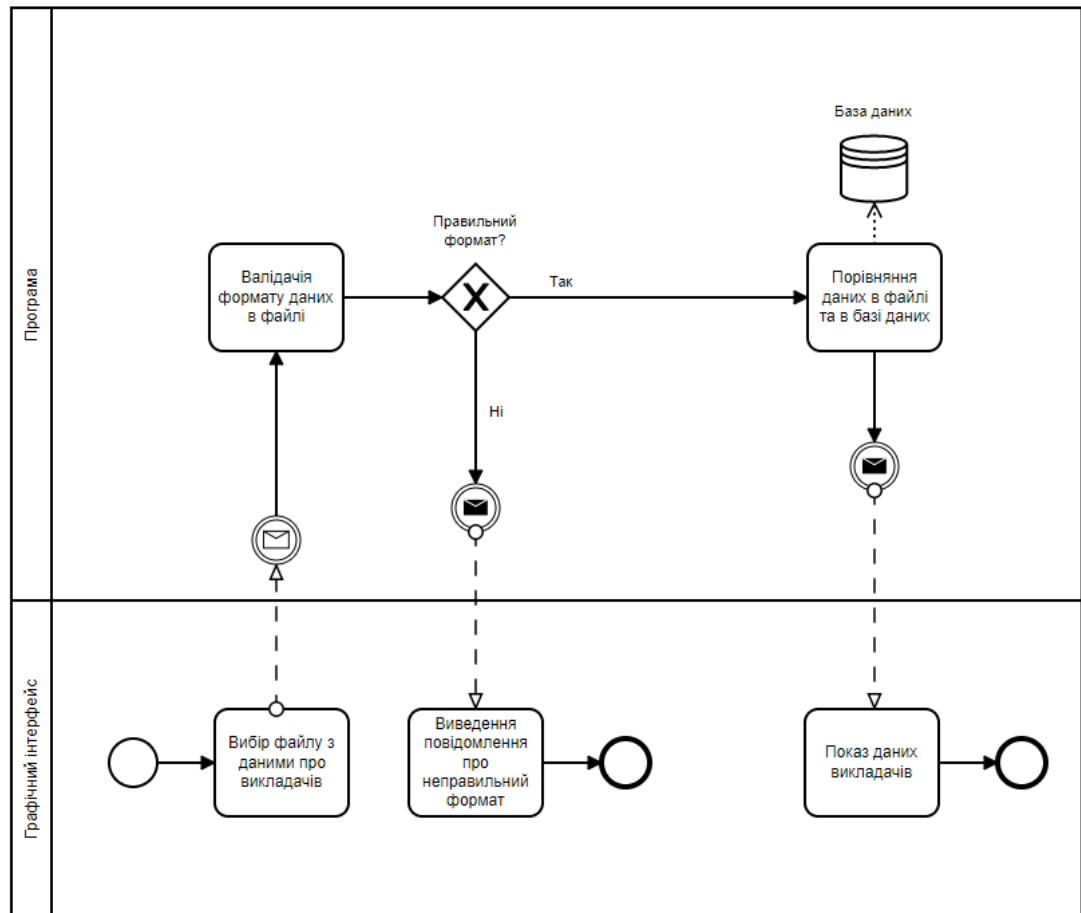


Рисунок 2.6 – Схема процесу збереження даних викладачів з веб-ресурсів

Опис процесу перевірки даних викладачів з файлу:

- користувач вибирає файл з даними про викладачів;
- система валідує формат даних в файлі;
- якщо валідація даних не успішна, система виводить повідомлення про неправильний формат;
- інакше система порівнює дані в файлі і в базі даних;
- далі система відображає результат перевірки.

## 2.2 Архітектура програмного забезпечення

Система складається з таких компонентів:

- Desktop Application – це XAML-розмітка для вікон додатку та класи обробки подій графічного інтерфейсу;

- Business logic – це класи, які описують бізнес логіку додатку та обробляють дані отримані з БД та веб-ресурсів;
- Web Socket – це бібліотека для отримання даних з веб-ресурсів Google Scholar, Web of Science, Scopus, Orcid;
- Database Library – це бібліотека для роботи з базою даних за допомогою Entity Framework.

Desktop Application описує вигляд вікон настільного додатку, та класи для обробки подій графічного інтерфейсу. Для створення графічного інтерфейсу була використана технологія Windows Presentation Foundation [6].

Business Logic надає та отримує дані з графічного інтерфейсу, реалізує взаємодію між базою даних та даними отриманими з веб-ресурсів, описує бізнес-логіку додатку.

В Web Socket описані парсери HTML-сторінок, проміжні сутності для збереження інформації з веб-сторінок та перетворення цих даних в сутності для бази даних. Для отримання веб-сторінок був використаний Selenium.WebDriver заснований на ChromeDriver [7]. Для отримання даних з HTML розмітки була використана бібліотека HTML Agility Pack [8].

В Database Library описані сутності бази даних, контекст БД, репозиторії для роботи з кожною таблицею та UnitOfWork для забезпечення єдиної точки доступу до БД. Для роботи з базою даних був використаний Entity Framework.

В якості СКБД для зберігання даних була обрана Microsoft SQL Server [9].

### 2.3 Опис структури бази даних

У таблицях 2.1 – 2.12 описані таблиці бази даних.

Таблиця Faculties призначена для зберігання даних про факультети.

Таблиця 2.1 – Faculties

Опис таблиці			
Поле	Опис	Тип даних	Ключ

## Продовження таблиці 2.1

FacultyId	Ідентифікатор факультету	int	Первинний
FacultyName	Назва факультету	nvarchar	

Таблиця Chairs призначена для зберігання даних про кафедри. За допомогою зовнішнього ключа FacultyId, таблиця Chairs пов'язана з таблицею Faculties зв'язком багато до одного.

## Таблиця 2.2 – Chairs

Опис таблиці			
Поле	Опис	Тип даних	Ключ
ChairId	Ідентифікатор кафедри	int	Первинний
FacultyId	Ідентифікатор факультету	int	Зовнішній
ChairName	Назва кафедри	nvarchar	

Таблиця Researchers призначена для зберігання даних про викладачів. За допомогою зовнішнього ключа ChairId, таблиця Researchers пов'язана з таблицею Chairs зв'язком багато до одного.

## Таблиця 2.3 – Researchers

Опис таблиці			
Поле	Опис	Тип даних	Ключ
ResearcherId	Ідентифікатор викладача	int	Первинний
FirstName	Ім'я викладача	nvarchar	

## Продовження таблиці 2.3

MiddleName	Ім'я по-батькові викладача	nvarchar	
LastName	Прізвище викладача	nvarchar	
ChairId	Ідентифікатор кафедри	int	Зовнішній

Таблиця ResearcherAccounts призначена для створення зв'язку багато до багатьох таблиць Researchers та Accounts.

## Таблиця 2.4 – ResearcherAccounts

Опис таблиці			
Поле	Опис	Тип даних	Ключ
ResearcherAccountId	Ідентифікатор	int	Первинний
ResearcherId	Ідентифікатор викладача	int	Зовнішній
AccountId	Ідентифікатор акаунту	int	Зовнішній

Таблиця Accounts призначена для зберігання даних про акаунти викладачів. За допомогою зовнішнього ключа TypeId, таблиця Accounts пов'язана з таблицею Types зв'язком багато до одного.

## Таблиця 2.5 – Accounts

Опис таблиці			
Поле	Опис	Тип даних	Ключ

Продовження таблиці 2.5

AccountId	Ідентифікатор акаунту	int	Первинний
TypeId	Ідентифікатор типу	int	Зовнішній
UserId	Ідентифікатор викладача	nvarchar	
FullName	Повне ім'я викладача	nvarchar	
HIndex	Індекс Гірша	int	
I10Index	I10 індекс	int	
PublicationsCount	Кількість публікацій	int	
CitationCount	Кількість цитувань	int	

Таблиця Types призначена для зберігання даних про типи аккаунти викладачів.

Таблиця 2.6 – Types

Опис таблиці			
Поле	Опис	Тип даних	Ключ
TypeId	Ідентифікатор типу	int	Первинний
TypeName	Назва типу	nvarchar	

Таблиця AccountPublications призначена для створення зв'язку багато до багатьох таблиць Researchers та Publications.

Таблиця 2.7 – AccountPublications

Опис таблиці			
Поле	Опис	Тип даних	Ключ
AccountPublicationId	Ідентифікатор	int	Первинний
AccountId	Ідентифікатор акаунту	int	Зовнішній
PublicationId	Ідентифікатор публікації	int	Зовнішній

Таблиця Publications призначена для зберігання даних про публікації аккаунтів.

Таблиця 2.8 – Publications

Опис таблиці			
Поле	Опис	Тип даних	Ключ
PublicationId	Ідентифікатор публікації	int	Первинний
Title	Назва публікації	nvarchar	
Source	Джерело публікації	nvarchar	
Date	Дата	datetime	
CitationCount	Кількість цитувань	int	
Authors	Автори	nvarchar	

Таблиця ResearcherOrcids призначена для створення зв'язку багато до багатьох таблиць Researchers та OrcidAccounts.

Таблиця 2.9 – ResearcherOrcids

Опис таблиці			
Поле	Опис	Тип даних	Ключ
ResearcherOrcidId	Ідентифікатор	int	Первинний
ResearcherId	Ідентифікатор викладача	int	Зовнішній
OrcidAccountId	Ідентифікатор акаунту Orcid	int	Зовнішній

Таблиця OrcidAccounts призначена для зберігання даних про акаунти Orcid.

Таблиця 2.10 – OrcidAccounts

Опис таблиці			
Поле	Опис	Тип даних	Ключ
OrcidAccountId	Ідентифікатор акаунту Orcid	int	Первинний
OrcidId	Ідентифікатор Orcid	nvarchar	
FullName	Повне ім'я викладача	nvarchar	
PublonsId	Ідентифікатор Web of Sciense	nvarchar	
ScopusAuthorId	Ідентифікатор Scopus	nvarchar	
PublonsLink	Посилання на Web of Sciense	nvarchar	
ScopusLink	Посилання на Scopus	nvarchar	

## Продовження таблиці 2.11

PublicationsCount	Кількість публікацій	int	
-------------------	----------------------	-----	--

Таблиця OrcidPublications призначена для зберігання даних про публікації аккаунтів Orcid. За допомогою зовнішнього ключа OrcidAccountId, таблиця OrcidPublications пов'язана з таблицею OrcidAccounts зв'язком багато до одного.

## Таблиця 2.12 – OrcidPublications

Опис таблиці			
Поле	Опис	Тип даних	Ключ
OrcidPublicationId	Ідентифікатор публікації	int	Первинний
OrcidAccountId	Ідентифікатор акаунту Orcid	int	Зовнішній
Title	Назва публікації	nvarchar	
Source	Джерело публікації	nvarchar	
Date	Дата	datetime	
Authors	Автори	nvarchar	

## 2.4 Конструювання програмного забезпечення

## Таблиця 2.13 – Опис класів Graphical User Interface

Клас	Опис
MainWindow	Клас що описує обробку подій основного вікна програми
FacultyWindow	Клас що описує обробку подій вікна для роботи з даними про факультети



Продовження таблиці 2.13

ChairWindow	Клас що описує обробку подій вікна для роботи з даними про кафедри
ResearcherWindow	Клас що описує обробку подій вікна для роботи з даними про викладачів
DialogWindow	Клас що описує обробку подій вікна для діалогів

Таблиця 2.14 – Опис інтерфейсів Business Logic

Клас	Опис
Index	Інтерфейс для всіх індексів, які ґрунтуються на кількості публікацій

Таблиця 2.15 – Опис класів Business Logic

Клас	Опис
FacultyBusiness	Клас що описує бізнес-логіку роботи з даними про факультети
ChairBusiness	Клас що описує бізнес-логіку роботи з даними про кафедри
ResearcherBusiness	Клас що описує бізнес-логіку роботи з даними про викладачів
FileBusiness	Клас що описує бізнес-логіку роботи з файлами а також збереження даних з веб-ресурсів
FacultyStatistics	Клас що описує бізнес-логіку роботи з статистикою по факультетах
ChairStatistics	Клас що описує бізнес-логіку роботи з статистикою по кафедрах

Продовження таблиці 2.15

ResearcherStatistics	Клас що описує бізнес-логіку роботи з статистикою по викладачах
HIndex	Клас що обчислює індекс Гірша з заданих публікацій, наслідує інтерфейс Index
I10Index	Клас що обчислює i10 індекс з заданих публікацій, наслідує інтерфейс Index
Faculty	Клас що зберігає дані про факультет, які потрібні для відображення та бізнес логіки
Chair	Клас що зберігає дані про кафедру, які потрібні для відображення та бізнес логіки
Researcher	Клас що зберігає дані про викладача, які потрібні для відображення та бізнес логіки
Data	Клас, що зберігає проміжні дані з файлів

Таблиця 2.16 – Опис класів Web Library

Клас	Опис
GoogleScholarParser	Клас що отримує дані про аккаунт викладача та всі його публікації з Google Scholar за заданим Id
ScopusParser	Клас що отримує дані про аккаунт викладача та всі його публікації з Scopus за заданим Id

Продовження таблиці 2.16

PublonsParser	Клас що отримує дані про аккаунт викладача та всі його публікації з Publons за заданим Id
OrcidParser	Клас що отримує дані про аккаунт викладача та всі його публікації з Orcid за заданим Id
Researcher	Клас для зберігання даних з веб-ресурсу про аккаунт викладача
Publication	Клас для зберігання даних з веб-ресурсу про публікацію
OrcidReasearcher	Клас для зберігання даних з веб-ресурсу Orcid про аккаунт викладача
OrcidPublication	Клас для зберігання даних з веб-ресурсу Orcid про публікацію
WebSocket	Клас для отримання WebDriver для роботи з веб-ресурсами

Таблиця 2.17 – Опис інтерфейсів Database Library

Клас	Опис
IRepository	Інтерфейс для репозиторію, сутність для роботи з однією таблицею із бази даних
IUnitOfWork	Інтерфейс для забезпечення єдиної точки доступу до бази даних

Таблиця 2.18 – Опис класів Database Library

Клас	Опис
Account	Клас, що описує сутність аккаунт з бази даних
AccountPublication	Клас, що описує сутність прив'язки публікації до аккаунту з бази даних
Chair	Клас, що описує сутність кафедра з бази даних
Faculty	Клас, що описує сутність факультет з бази даних
OrcidAccount	Клас, що описує сутність Orcid аккаунт з бази даних
OrcidPublication	Клас, що описує сутність Orcid публікація з бази даних
Publication	Клас, що описує сутність публікація з бази даних
Researcher	Клас, що описує сутність викладач з бази даних
ResearcherAccount	Клас, що описує сутність прив'язки аккаунту до викладача з бази даних
ResearcherOrcid	Клас що описує сутність прив'язки Orcid аккаунту до викладача з бази даних
Type	Клас що описує сутність тип аккаунту з бази даних
KpiResearcherContext	Клас що описує контекст бази даних
AccountPuplicationRepository	Клас що описує роботу з таблицею прив'язок публікацій до аккаунтів, наслідує інтерфейс IRepository

## Продовження таблиці 2.18

AccountRepository	Клас що описує роботу з таблицею аккаунтів, наслідує інтерфейс IRepository
ChairRepository	Клас що описує роботу з таблицею кафедр, наслідує інтерфейс IRepository
FacultyRepository	Клас що описує роботу з таблицею факультетів, наслідує інтерфейс IRepository
OrcidAccountRepository	Клас що описує роботу з таблицею Orcid аккаунтів, наслідує інтерфейс IRepository
OrcidPublicationRepository	Клас що описує роботу з таблицею Orcid публікацій, наслідує інтерфейс IRepository
PublicationRepository	Клас що описує роботу з таблицею публікацій, наслідує інтерфейс IRepository
ResearcherAccountRepository	Клас що описує роботу з таблицею прив'язок аккаунтів до викладачів, наслідує інтерфейс IRepository
ResearcherOrcidRepository	Клас що описує роботу з таблицею прив'язок Orcid аккаунтів до викладачів, наслідує інтерфейс IRepository
ResearcherRepository	Клас що описує роботу з таблицею викладачів, наслідує інтерфейс IRepository

## Продовження таблиці 2.18

TypeRepository	Клас що описує роботу з таблицею типами аккаунтів, наслідує інтерфейс IRepository
EUnitOfWork	Клас що дає доступ до репозиторіїв, наслідує інтерфейс IUnitOfWork

Розглянемо основні функції та методи класів та інтерфейсів нашого програмного забезпечення.

Таблиця 2.19 – Опис методів класів та інтерфейсів Business Logic

Клас/Інтерфейс	Метод	Опис
Index	GetIndex()	Отримання індексу з списку цитувань
HIndex	GetIndex()	Реалізація методу інтерфейсу Index, отримання індексу Гірша з списку цитувань
I10Index	GetIndex()	Реалізація методу інтерфейсу Index, отримання i10 індексу з списку цитувань
FacultyBusiness	GetFaculties()	Отримання списку всіх факультетів із бази даних без детальної інформації

## Продовження таблиці 2.19

FacultyBusiness	GetDetailFaculties()	Отримання списку всіх факультетів із бази даних із індексом Гірша та i10 індексом
FacultyBusiness	CreateFaculty(Faculty faculty)	Збереження факультету в базу даних
FacultyBusiness	EditFaculty(Faculty faculty)	Оновлення даних про факультет в базі даних
FacultyBusiness	DeleteFaculty(int facultyId)	Видалення факультету з бази даних
ChairBusiness	GetChairs()	Отримання списку всіх кафедр із бази даних без детальної інформації
ChairBusiness	GetDetailChairs()	Отримання списку всіх кафедр із бази даних із індексом Гірша та i10 індексом
ChairBusiness	GetDetailChairs(int facultyId)	Отримання списку кафедр заданого факультету із бази даних із індексом Гірша та i10 індексом
ChairBusiness	GetChair(int chairId)	Отримання даних про кафедру за заданим ідентифікатором із бази даних із індексом Гірша та i10 індексом

## Продовження таблиці 2.19

ChairBusiness	CreateChair(Chair chair)	Збереження кафедри в базу даних
ChairBusiness	EditChair(Chair chair)	Оновлення даних про кафедру в базі даних
ChairBusiness	DeleteChair(int chairId)	Видалення кафедри з бази даних
ResearcherBusiness	GetResearchers()	Отримання списку всіх викладачів із бази даних без детальної інформації
ResearcherBusiness	GetResearchers(int chairId)	Отримання списку викладачів заданої кафедри із бази даних із індексом Гірша та i10 індексом
ResearcherBusiness	GetResearchers(string firstName, string middleName, string lastName)	Отримання списку викладачів із бази даних із заданим прізвищем, ім'ям та ім'ям по-батькові
ResearcherBusiness	GetResearcher(int researcherId)	Отримання даних про викладача за заданим ідентифікатором із бази даних із індексом Гірша та i10 індексом
ResearcherBusiness	CreateResearcher(Researcher researcher)	Збереження викладача в базу даних



## Продовження таблиці 2.19

ResearcherBusiness	EditResearcher( Researcher researcher)	Оновлення даних про викладача в базі даних
ResearcherBusiness	DeleteResearcher(int researcherId)	Видалення викладача з бази даних
ResearcherBusiness	SaveResearcher( WebLibrary.Entities .Researcher researcher)	Збереження даних про викладача, його аккаунти та публікації в базу даних
FileBusiness	SaveFile(string fileName)	Завантаження даних про викладачів, які описані в файлі, із Orcid, Google Scholar, Web of Science, Scopus, та подальше збереження цих даних в базу даних
FileBusiness	CheckFile(string fileName)	Перевірка коректності індексу Гірша для викладачів, які описані в файлі
FileBusiness	CheckData(Data data)	Перевірка запису викладача на коректні дані про індекс Гірша та i10 індекс
FileBusiness	SaveOrcidAccount( OrcidAccount account)	Збереження облікового запису Orcid та його публікацій в базу даних

## Продовження таблиці 2.19

FileBusiness	SaveData(Data data)	Завантаження даних про викладача із Orcid, Google Scholar, Web of Science, Scopus, та подальше збереження цих даних в базу даних
FacultyStatistics	GetFacultyHIndex GoogleScholarStatistics()	Отримання статистики індексу Гірша в Google Scholar по всіх факультетам
FacultyStatistics	GetFacultyHIndex PublonsStatistics()	Отримання статистики індексу Гірша в Publons по всіх факультетам
FacultyStatistics	GetFacultyI10Index GoogleScholarStatistics()	Отримання статистики i10 індексу в Google Scholar по всіх факультетам
FacultyStatistics	GetFacultyI10Index PublonsStatistics()	Отримання статистики i10 індексу в Publons по всіх факультетам
FacultyStatistics	InitializeFaculty Statistics()	Створення всіх статистик факультетів для того щоб закешувати дані статистик

Продовження таблиці 2.19

FacultyStatistics	GetChairStatistics( int facultyId, int typeId, int index)	Отримання статистики по кафедрам обраного факультету, обраної наукометричної бази даних та обраного показника
FacultyStatistics	GetResearcherStatistics( int chairId, int typeId, int index)	Отримання статистики по викладачам обраного факультету, обраної наукометричної бази даних та обраного показника

Таблиця 2.20 – Опис методів класів та інтерфейсів Web Library

Клас/Інтерфейс	Метод	Опис
GoogleScholarParser	GetResearcher(string userId)	Завантаження даних про обліковий запис викладача Google Scholar та всіх його публікацій
PublonsParser	GetResearcher(string userId)	Завантаження даних про обліковий запис викладача Publons та всіх його публікацій
ScopusParser	GetResearcher(string userId)	Завантаження даних про обліковий запис викладача Scopus та всіх його публікацій

## Продовження таблиці 2.20

OrcidParser	GetResearcher(string userId)	Завантаження даних про обліковий запис викладача Orcid та всіх його публікацій
WebSocket	GetPage(string url)	Завантаження веб- сторінки за заданим URL адресом

Таблиця 2.21 – Опис методів класів та інтерфейсів Database Library

Клас/Інтерфейс	Метод	Опис
IRepository	GetAll()	Отримання всіх записів із бази даних заданого типу
IRepository	Get(int id)	Отримання одного запису заданого типу за заданим ідентифікатором із бази даних
IRepository	Find(Func<T, Boolean> predicate)	Отримання всіх записів заданого типу, які відповідають заданому критерію відбору, з бази даних
IRepository	Create(T item)	Створення запису заданого типу в базі даних

## Продовження таблиці 2.21

IRepository	Update(T item)	Оновлення запису заданого типу в базі даних
IRepository	Delete(int id)	Видалення запису заданого типу за заданим ідентифікатором в базі даних
IUnitOfWork	Accounts	Отримання репозиторію для роботи з таблицею облікових записів викладачів
IUnitOfWork	AccountPublications	Отримання репозиторію для роботи з таблицею прив'язок публікацій до облікових записів
IUnitOfWork	Chairs	Отримання репозиторію для роботи з таблицею кафедр
IUnitOfWork	Faculties	Отримання репозиторію для роботи з таблицею факультетів
IUnitOfWork	OrcidAccounts	Отримання репозиторію для роботи з таблицею облікових записів Orcid

## Продовження таблиці 2.21

IUnitOfWork	OrcidPublications	Отримання репозиторію для роботи з таблицею публікацій облікових записів Orcid
IUnitOfWork	Publications	Отримання репозиторію для роботи з таблицею публікацій облікових записів викладачів
IUnitOfWork	Researchers	Отримання репозиторію для роботи з таблицею викладачів
IUnitOfWork	ResearcherAccounts	Отримання репозиторію для роботи з таблицею прив'язок викладачів до облікових записів
IUnitOfWork	ResearcherOrcids	Отримання репозиторію для роботи з таблицею прив'язок викладачів до облікових записів Orcid
IUnitOfWork	Types	Отримання репозиторію для роботи з таблицею типів облікових записів
IUnitOfWork	Save()	Збереження всіх змін в базу даних

## 2.5 Аналіз безпеки даних

Оскільки всі дані які, зберігаються в базі даних, є публічними нам не потрібно забезпечувати безпеку цих даних.

## 2.6 Висновки по розділу

В даному розділі було створено діаграми бізнес процесів взаємодії користувача з розроблюваним додатком, також була розроблена архітектура програмного продукту. Також були розглянуті технології, які будуть використовуватись для розробки, а саме мова програмування C#, СКБД MS SQL Server, фреймворк WPF для графічного інтерфейсу та Selenium.ChromeDriver для роботи з веб-ресурсами. Також було проаналізована безпеку даних в системі.

### 3 АНАЛІЗ ЯКОСТІ ТА ТЕСТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

#### 3.1 Аналіз якості ПЗ

Тестування програмного забезпечення є обов'язковим етапом його розробки, адже воно гарантує необхідний рівень якості, відповідність вимогам до програмного продукту, наявність всього заявленого функціоналу.

Тестування програмного забезпечення можна розділити на дві великі категорії: ручне тестування та автоматизоване.

Автоматизоване тестування спрощує процес розробки програмного продукту, адже воно гарантує, що зміни в модулі не призвели до припинення роботи вже працюючого функціоналу. Але розробка автоматизованого тестування потребує часу.

Ручне тестування зазвичай використовується для складних процесів, які потрібно виконати 1-3 рази, наприклад тестування графічного інтерфейсу, адже час на розробку автоматизованого тесту значно перевищує час виконання ручного тесту.

При розробці настільних додатків, тестування є дуже важливим адже користувачі дуже часто допускають помилки при введенні даних, і тому без тестування не можна отримати високу ступінь надійності системи до відмов через неправильні дані.

Будуть протестовані наступні компоненти програмного забезпечення:

- створення записів про факультет, кафедру, викладача;
- редагування записів про факультет, кафедру, викладача;
- видалення записів про факультет, кафедру, викладача;
- отримання даних з веб-ресурсів;
- робота інтерфейсу користувача.



### 3.2 Опис процесів тестування

Для тестування програмного продукту я використаю такі підходи:

- тестування графічного інтерфейсу;
- модульне тестування;
- інтеграційне тестування.

За допомогою тестування графічного інтерфейсу я перевірю як система реагує на некоректно введені дані, а також як вона працює з валідними даними.

За допомогою модульного тестування буде перевірятись робота окремих компонентів бізнес-логіки, компонентів для роботи з базою даних та веб-ресурсами.

За допомогою інтеграційного тестування буде перевірятись взаємодія між графічним інтерфейсом, бізнес-логікою та компонентами для роботи з базою даних та веб-ресурсами.

### 3.3 Опис контрольного прикладу

В якості контрольного прикладу буде розглянуто тестування отримання даних з веб-ресурсів. Даний приклад наведено у таблиці 3.1.

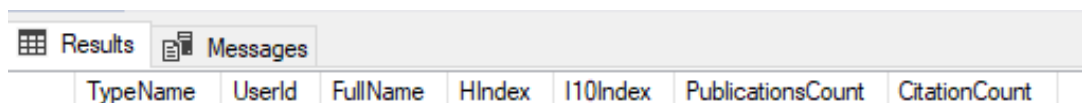
Таблиця 3.1 – Опис контрольного прикладу

Мета тесту	Перевірка коректності збереження даних з веб-ресурсів
Початковий стан	БД не містить записів викладачів
Вхідні дані	test;test;Gorshkov;Vyacheslav;M;0000-0002-7700-5649;0
Опис процесу тестування	Запустити функцію для збереження даних викладача з веб-ресурсів
Кінцевий результат	БД містить інформацію про викладача з веб-ресурсу

## Продовження таблиці 3.1

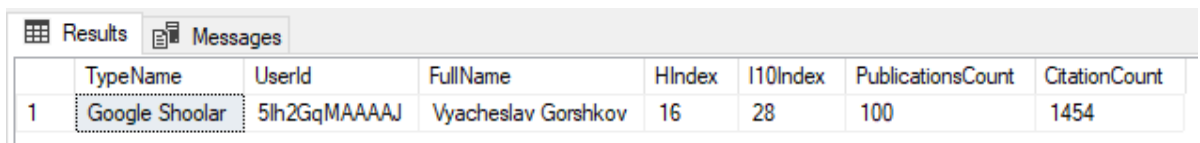
Очікуваний результат	БД буде містити інформацію про викладача з веб-ресурсу
----------------------	--

Розглянутий тест перевіряє базову функціональність та виконується без графічного інтерфейсу в командному рядку. Для виконання тесту треба перевірити базу даних на відсутність записів, потім запустити відповідний консольний додаток. Після цього слід перевірити базу даних на наявність запису. Результат виконання тесту наведеного в таблиці 3.1 зображено на рисунках 3.1 та 3.2.



TypeName	UserId	FullName	HIndex	I10Index	PublicationsCount	CitationCount
----------	--------	----------	--------	----------	-------------------	---------------

Рисунок 3.1 – База даних до контрольного тесту



TypeName	UserId	FullName	HIndex	I10Index	PublicationsCount	CitationCount
Google Shoolar	5lh2GqMAAAAJ	Vyacheslav Gorshkov	16	28	100	1454

Рисунок 3.2 – База даних після контрольного тесту

## 3.4 Висновок до розділу

В даному розділі були описані основні компоненти, які необхідно протестувати для коректної роботи програми та підходи до тестування цих компонентів.

Також наведений контрольний приклад в розгорнутому вигляді для перевірки коректності збереження даних з веб-ресурсів.

## 4 ВПРОВАДЖЕННЯ ТА СУПРОВІД ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

### 4.1 Розгортання програмного забезпечення

На Рисунку 4.1 зображена діаграма розгортання програмного забезпечення.

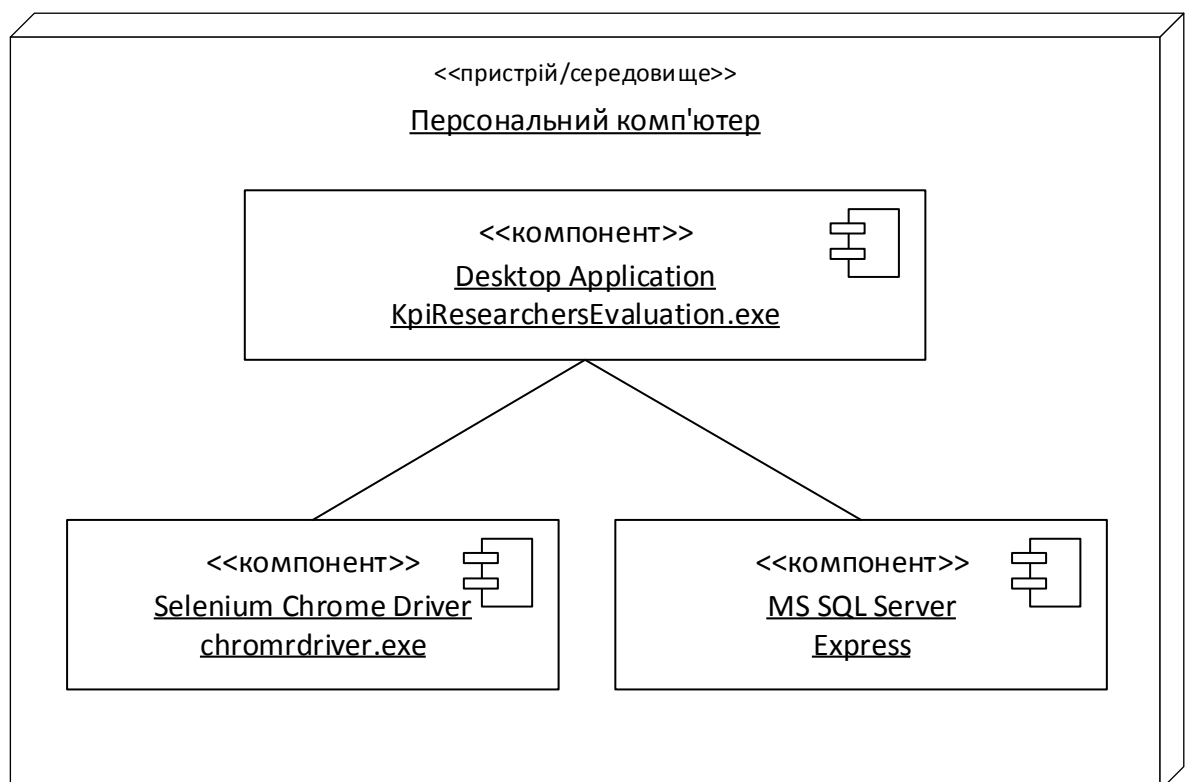


Рисунок 4.1 – Діаграма розгортання

Для розгортання проекту потрібно:

- встановити СКБД MS SQL Server;
- створити базу даних;
- встановити настільний додаток KpiResearchersEvaluation.

#### 4.1.1 Встановлення СКБД MS SQL Server

Для встановлення системи керування базами даних необхідно перейти за наступним посиланням <https://www.microsoft.com/ru-ru/sql-server/sql-server-editions-express> та завантажити інсталяційний файл. Далі запустити інсталяційний файл і виконати встановлення MS SQL Server.

#### 4.1.2 Створення бази даних

Для того, щоб створити базу даних необхідно запустити скрипт CreateDB.sql в системі керування базами даних.

#### 4.1.3 Встановлення настільного додатку KpiResearchersEvaluation

Для того, щоб встановити настільний додаток необхідно запустити інсталяційний файл setup.exe

#### 4.2 Робота з програмним забезпеченням

Інструкція наведена в додатку Інструкція користувача

## ВИСНОВКИ

У ході дипломного проекту був проведений ґрунтовний аналіз предметної області, також були виявлені основні конкуренти із схожим функціоналом, були проаналізовані їх основні переваги та недоліки.

Було проведене моделювання програмного забезпечення, створені функціональні та нефункціональні вимоги. Для моделювання бізнес-процесів застосунку були створені BPMN діаграми. Також була описана архітектура програмного забезпечення.

Для розробки програмного забезпечення була використана мова С#, інтегроване середовище розробки Visual Studio 2019 Community та бібліотеки Selenium.ChromeDriver, HTML Agility Pack, Entity Framework.

Також була створена модель бази даних, яка дає змогу ефективно та надійно здійснювати доступ до даних, що надходять та використовуються в оцінюванні результатів наукової діяльності викладачів. Для керування базою даних обрана СКБД Microsoft SQL Server Express.

Була розроблена детальна інструкція користувача та схема розгортання програмного продукту.

Під час розробки програмного забезпечення були застосовані набуті знання з архітектури програмного забезпечення, архітектури баз даних, розробки настільних додатків, об'єктно-орієнтованого програмування.

Був створений настільний додаток для вирішення поставленої задачі.

					КПІ.ІП-5104.045490.01.81	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		61

## ПЕРЕЛІК ПОСИЛАНЬ

- 1) Google Scholar [Електронний ресурс] – Режим доступу:  
<https://scholar.google.com.ua/>.
- 2) Scopus [Електронний ресурс] – Режим доступу:  
<https://www.scopus.com/>.
- 3) Web of Science [Електронний ресурс] – Режим доступу:  
<https://publons.com/>.
- 4) Індекс Гірша [Електронний ресурс] – Режим доступу:  
<https://uk.wikipedia.org/wiki/H-%D1%96%D0%BD%D0%B4%D0%B5%D0%BA%D1%81>.
- 5) Orcid [Електронний ресурс]. – Режим доступу: <https://orcid.org/>.
- 6) Windows Presentation Foundation [Електронний ресурс] – Режим доступу: [https://uk.wikipedia.org/wiki/Windows\\_Presentation\\_Foundation](https://uk.wikipedia.org/wiki/Windows_Presentation_Foundation).
- 7) ChromeDriver [Електронний ресурс] – Режим доступу:  
<http://chromedriver.chromium.org/downloads>.
- 8) HTML Agility Pack [Електронний ресурс] – Режим доступу:  
<https://html-agility-pack.net/>.
- 9) Microsoft SQL Server [Електронний ресурс] – Режим доступу:  
[https://uk.wikipedia.org/wiki/Microsoft\\_SQL\\_Server](https://uk.wikipedia.org/wiki/Microsoft_SQL_Server).
- 10) MacDonald M. Pro WPF 4.5 in C#: Windows Presentation Foundation in .NET 4.5 / Matthew MacDonald. – New York: Apress, 2012. – 1078 с. – (4th edition).
- 11) Albahari J. C# 7.0 in a Nutshell: The Definitive Reference / J. Albahari, B. Albahari. – Sebastopol, California: O'Reilly Media, 2017. – 1088 с. – (1st edition).
- 12) Richter J. CLR via C# / Jeffrey Richter. – New York: Microsoft Press, 2012. – 896 с. – (4th edition). – (Developer Reference).
- 13) Ben-Gan I. T-SQL Fundamentals / Itzik Ben-Gan. – New York: Microsoft Press, 2016. – 464 с. – (3rd edition).

					КП.ІП-5104.045490.01.81	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		62

14) Lerman J. Programming Entity Framework: DbContext / J. Lerman, R. Miller. – Sebastopol, California: O'Reilly Media, 2012. – 258 с. – (1st edition).

					КПІ.ІП-5104.045490.01.81	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		63

**Факультет інформатики та обчислювальної техніки**  
**Кафедра автоматизованих систем обробки інформації і управління**

**“ЗАТВЕРДЖЕНО”**

В.о. завідувача кафедри

\_\_\_\_\_ О.А. Павлов

“ \_\_\_\_ ” \_\_\_\_\_ 2019 р.

**Програмне забезпечення оцінювання результатів наукової**

**діяльності викладачів**

**Технічне завдання**

КП.ІП-5104.045490.02.91

**“ПОГОДЖЕНО”**

Керівник проекту:

\_\_\_\_\_ О.Д. Фіногенов

Нормоконтроль:

\_\_\_\_\_ Ліщук К.І.

Виконавець:

\_\_\_\_\_ В.Ю. Гладишко

Київ – 2019 року



## ЗМІСТ

<b>1</b>	<b>НАЙМЕНУВАННЯ ТА ГАЛУЗЬ ЗАСТОСУВАННЯ.....</b>	<b>3</b>
<b>2</b>	<b>ПІДСТАВА ДЛЯ РОЗРОБКИ.....</b>	<b>4</b>
<b>3</b>	<b>ПРИЗНАЧЕННЯ РОЗРОБКИ .....</b>	<b>5</b>
<b>4</b>	<b>ВИМОГИ ДО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ .....</b>	<b>6</b>
4.1	Вимоги до функціональних характеристик.....	6
4.2	Вимоги до надійності .....	6
4.3	Умови експлуатації .....	6
4.4	Вимоги до складу і параметрів технічних засобів .....	7
4.5	Вимоги до інформаційної та програмної сумісності .....	7
4.6	Вимоги до маркування та пакування.....	7
4.7	Вимоги до транспортування та зберігання .....	7
4.8	Спеціальні вимоги.....	7
<b>5</b>	<b>ВИМОГИ ДО ПРОГРАМНОЇ ДОКУМЕНТАЦІЇ .....</b>	<b>8</b>
<b>6</b>	<b>СТАДІЇ І ЕТАПИ РОЗРОБКИ.....</b>	<b>9</b>
<b>7</b>	<b>ПОРЯДОК КОНТРОЛЮ ТА ПРИЙМАННЯ.....</b>	<b>10</b>

## 1 НАЙМЕНУВАННЯ ТА ГАЛУЗЬ ЗАСТОСУВАННЯ

### Назва розробки:

«Програмне забезпечення оцінювання результатів наукової діяльності викладачів»

### Галузь застосування:

Наведене технічне завдання поширюється на розробку додатка «Програмне забезпечення оцінювання результатів наукової діяльності викладачів»[045490], котра використовується для збору результатів наукової діяльності викладачів на основі бібліографічних баз даних та призначена для використання в вищих навчальних закладах для оцінювання наукової діяльності.

					КПІ.ІП-5104.045490.02.91	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		3

## 2 ПІДСТАВА ДЛЯ РОЗРОБКИ

Підставою для розробки «Програмне забезпечення оцінювання результатів наукової діяльності викладачів» є завдання на дипломне проектування, затверджене кафедрою автоматизованих систем обробки інформації і управління Національного технічного університету України «Київський політехнічний інститут імені Ігоря Сікорського» (КПІ ім.Ігоря Сікорського).

					КПІ.ІП-5104.045490.02.91	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		4

### 3 ПРИЗНАЧЕННЯ РОЗРОБКИ

Розробка призначена для оцінювання результатів наукової діяльності викладачів на основі бібліографічних баз даних.

Метою розробки є:

- автоматизація збору даних із ресурсів Google Scholar, Scopus та Web of Science;
- визначення результатів наукової діяльності кафедр та факультетів, що представлені в бібліографічних базах даних.

					КПІ.ІП-5104.045490.02.91	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		5

## 4 ВИМОГИ ДО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

### 4.1 Вимоги до функціональних характеристик

4.1.1 Програмне забезпечення повинно забезпечувати виконання наступних основних функцій:

#### 4.1.1.1 Для користувача:

- додавання, редагування, перегляд інформації о викладачах, кафедрах, факультетах;
- перегляд результатів наукової діяльності викладача;
- перегляд результатів наукової діяльності кафедри;
- перегляд результатів наукової діяльності факультету.

### 4.1.2 Розробку виконати на платформі Windows.

### 4.1.3 Додаткові вимоги:

- можливість оновлення даних в базі за встановленим розкладом.

### 4.2 Вимоги до надійності

#### 4.2.1 Передбачити контроль введення інформації.

#### 4.2.2 Передбачити захист від некоректних дій користувача.

#### 4.2.3 Забезпечити цілісність інформації в базі даних.

### 4.3 Умови експлуатації

#### 4.3.1 Умови експлуатації згідно СанПін 2.2.2.542 – 96.

#### 4.3.2 Обслуговування

Розроблюване програмне забезпечення функціонує в автоматичному режимі та не вимагає обслуговування.

					КПІ.ІП-5104.045490.02.91	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		6

#### 4.4 Вимоги до складу і параметрів технічних засобів

4.4.1 Програмне забезпечення повинно функціонувати на IBM-сумісних персональних комп'ютерах.

#### 4.4.2 Мінімальна конфігурація технічних засобів:

- центральний процесор: Intel Core 2 або AMD Athlon 64 (2.0 GHz, 2 cores)
- оперативна пам'ять: 2 GB DDR2
- постійна пам'ять: 64 GB.

#### 4.5 Вимоги до інформаційної та програмної сумісності

4.5.1 Програмне забезпечення повинно працювати під управлінням операційних систем сімейства WIN32 (Windows XP, Windows 7, Windows 10).

4.5.2 Вхідні дані повинні бути представлені в наступному форматі: ORCID– ідентифікатор викладача, ПІБ, місце роботи.

4.5.3 Результати повинні бути представлені в наступному форматі: \*.xml, \*.csv.

4.5.4 Програмне забезпечення повинно бути розроблено на мові програмування C# з використанням фреймворків WPF та Entity Framework в IDE Visual Studio.

#### 4.6 Вимоги до маркування та пакування

Вимоги до маркування та пакування не пред'являються.

#### 4.7 Вимоги до транспортування та зберігання

Вимоги до транспортування та зберігання не пред'являються.

#### 4.8 Спеціальні вимоги

Згенерувати установчу версію програмного забезпечення.

					КПІ.ІП-5104.045490.02.91	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		7

## 5 ВИМОГИ ДО ПРОГРАМНОЇ ДОКУМЕНТАЦІЇ

5.1 Програмні модулі, котрі розробляються, повинні бути задокументовані, тобто тексти програм повинні містити всі необхідні коментарі.

5.2 Програмне забезпечення повинно мати довідникову систему

5.3 У склад супроводжувальної документації повинні входити наступні документи:

5.3.1 Пояснювальна записка не менше ніж на 50 аркушах формату А4 (без додатків 5.3.2 - 5.3.4).

5.3.2 Технічне завдання.

5.3.3 Опис програми.

5.3.4 Керівництво користувача.

5.3.5 Програма та методика тестування.

5.4 Графічна частина повинна бути виконана на 3х листах формату А3, котрі включаються у якості додатків до пояснювальної записки:

5.4.1 Схема бази даних.

5.4.2 Схема структурна класів програмного забезпечення.

5.4.3 Креслення вигляду екранних форм.

## 6 СТАДІЇ І ЕТАПИ РОЗРОБКИ

	Назва етапу	Строк	Звітність
1	Вивчення літератури за тематикою проекту	20.04.19	
2	Розробка технічного завдання	23.04.19	Технічне завдання
3	Аналіз вимог та уточнення специфікацій	25.04.19	Специфікації програмного забезпечення
4	Проектування структури програмного забезпечення, проектування компонентів	28.04.19	Схема структурна програмного забезпечення та специфікація компонентів
5	Програмна реалізація програмного забезпечення	20.05.19	Вихідні тексти програмного забезпечення
6	Тестування програмного забезпечення	27.05.19	Тести, результати тестування
7	Розробка матеріалів текстової частини проекту	30.05.19	Пояснювальна записка.
8	Розробка матеріалів графічної частини проекту	07.06.19	Графічний матеріал проекту
9	Оформлення технічної документації проекту	09.06.19	Технічна документація



## 7 ПОРЯДОК КОНТРОЛЮ ТА ПРИЙМАННЯ

### 7.1 Види випробувань

Тестування розробленого програмного продукту виконується відповідно до “Програми та методики тестування”.

					КПІ.ІП-5104.045490.02.91	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		10

**Факультет інформатики та обчислювальної техніки**  
**Кафедра автоматизованих систем обробки інформації і управління**

**“ЗАТВЕРДЖЕНО”**

**В.о. завідувача кафедри**

\_\_\_\_\_ **О.А. Павлов**

**“     ”     \_\_\_\_\_ 2019 р.**

**Програмне забезпечення оцінювання результатів наукової**  
**діяльності викладачів**

**Програма та методика тестування**

**КП.ІІ-5104.045490.03.51**

**“ПОГОДЖЕНО”**

**Керівник проекту:**

\_\_\_\_\_ **О.Д. Фіногенов**

**Нормоконтроль:**

\_\_\_\_\_ **Ліщук К.І.**

**Виконавець:**

\_\_\_\_\_ **В.Ю. Гладишко**

**Київ – 2019 року**

## ЗМІСТ

1	НАЙМЕНУВАННЯ ТА ГАЛУЗЬ ЗАСТОСУВАННЯ.....	3
2	МЕТА ТЕСТУВАННЯ .....	3
3	МЕТОДИ ТЕСТУВАННЯ .....	3
4	ЗАСОБИ ТА ПОРЯДОК ТЕСТУВАННЯ.....	4

## 1 НАЙМЕНУВАННЯ ТА ГАЛУЗЬ ЗАСТОСУВАННЯ

Програмне забезпечення оцінювання результатів наукової діяльності викладачів являє собою настільний додаток, створений за допомогою мови програмування C#, з використанням фреймворків Entity Framework, Windows Presentation Foundation, з використанням бібліотек HTML Agility Pack, Selenium Chrome Driver, Newtonsoft Json, також при розробці використовувався XAML та Transact-SQL.

## 2 МЕТА ТЕСТУВАННЯ

Будуть протестовані наступні компоненти програмного забезпечення:

- створення записів про факультет, кафедру, викладача;
- редагування записів про факультет, кафедру, викладача;
- видалення записів про факультет, кафедру, викладача;
- отримання даних з веб-ресурсів;
- робота інтерфейсу користувача.

## 3 МЕТОДИ ТЕСТУВАННЯ

Для тестування програмного продукту я використаю такі підходи:

- тестування графічного інтерфейсу;
- модульне тестування.

За допомогою тестування графічного інтерфейсу я перевірю як система опрацьовує правильні дані, а також як вона реагує на некоректно введені дані.

За допомогою модульного тестування буде перевірятись робота окремих компонентів для роботи з базою даних, компонентів для роботи з веб-ресурсами та компонентів бізнес-логіки.

					КПІ.ІП-5104.045490.03.51	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		3

#### 4 ЗАСОБИ ТА ПОРЯДОК ТЕСТУВАННЯ

Тестування буде проводитись за допомогою інструментарію NUnit 3.

Працездатність настільного додатку буде проводитись шляхом:

- ручного тестування – введення граничних та недопустимих значень в текстові поля графічного інтерфейсу користувача;
- модульного тестування компонентів для взаємодії з базою даних;
- модульного тестування компонентів для взаємодії з веб-ресурсами;
- модульного тестування компонентів бізнес логіки;
- ручного тестування на відповідність функціональним вимогам;
- тестування зручності використання;
- тестування інтерфейсу.

					КПІ.ІП-5104.045490.03.51	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		4

**Факультет інформатики та обчислювальної техніки**  
**Кафедра автоматизованих систем обробки інформації і управління**

**“ЗАТВЕРДЖЕНО”**

В.о. завідувача кафедри

\_\_\_\_\_ О.А. Павлов

“ \_\_\_\_ ” \_\_\_\_\_ 2019 р.

**Програмне забезпечення оцінювання результатів наукової**  
**діяльності викладачів**

**Опис програми**

КПІ.ІІ-5104.045490.04.13

**“ПОГОДЖЕНО”**

Керівник проекту:

\_\_\_\_\_ О.Д. Фіногенов

Нормоконтроль:

\_\_\_\_\_ Ліщук К.І.

Виконавець:

\_\_\_\_\_ В.Ю. Гладишко

Київ – 2019 року

***Тексти програмного коду***  
***Програмне забезпечення оцінювання результатів***  
***наукової діяльності викладачів***

(Найменування програми (документа))

*DVD-R*

(Вид носія даних)

*14 арк, 112 877 Кб*

(Обсяг програми (документа) , арк.,) Кб)

Київ – 2019

					КПІ.ІП-5104.045490.04.13	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		2

```

using BusinessLogic.Entities;
using BusinessLogic.Indexes;
using DatabaseLibrary.DatabaseModel;
using DatabaseLibrary.Repositories;
using OpenQA.Selenium;
using System.Collections.Generic;
using System.IO;
using System.Linq;
using System.Threading.Tasks;
using WebLibrary;
using WebLibrary.Parsers;
using OrcidAccount = WebLibrary.Entities.OrcidAccount;
using OrcidPublication = WebLibrary.Entities.OrcidPublication;

namespace BusinessLogic.Business
{
    public static class FileBusiness
    {
        private static EFUnitOfWork unitOfWork = new EFUnitOfWork();

        public static IEnumerable<Data> SaveFile(string fileName)
        {
            var list = new List<Data>();
            var dataRaw = File.ReadAllLines(fileName).Select(line =>
line.Split(';'));
            foreach (var row in dataRaw)
            {
                var data = new Data
                {

```



```

Faculty = row[0],
Chair = row[1],
LastName = row[2],
FirstName = row[3],
MiddleName = row[4],
Orcid = row[5],
GoogleHIndexOld = int.Parse(row[6])
};

```

```

SaveData(data);
data = CheckData(data);

```

```

list.Add(data);
}
return list;
}

```

```

public static IEnumerable<Data> CheckFile(string fileName)
{
    var list = new List<Data>();
    var dataRaw = File.ReadAllLines(fileName).Select(line =>
line.Split(';'));
    foreach (var row in dataRaw)
    {
        var data = new Data
        {
            Faculty = row[0],
            Chair = row[1],
            LastName = row[2],

```

```

        FirstName = row[3],
        MiddleName = row[4],
        Orcid = row[5],
        GoogleHIndexOld = int.Parse(row[6])
    };

```

```

data = CheckData(data);

```

```

        list.Add(data);
    }
    return list;
}

```

```

public static Data CheckData(Data data)
{
    var researchers = unitOfWork.Researchers.Find(item =>
        item.FirstName.ToLower() == data.FirstName.ToLower()
        && item.MiddleName.ToLower() ==
data.MiddleName.ToLower()
        && item.LastName.ToLower() == data.LastName.ToLower()
        && item.Chair.ChairName.ToLower() == data.Chair.ToLower()
        && item.Chair.Faculty.FacultyName.ToLower() ==
data.Faculty.ToLower());

    if (researchers.Count() == 0)
    {
        SaveData(data);
        data.GoogleHIndex = new
Entities.Researcher(unitOfWork.Researchers.Find(item =>

```

```

        item.FirstName.ToLower() == data.FirstName.ToLower()
        && item.MiddleName.ToLower() == data.MiddleName.ToLower()
        && item.LastName.ToLower() == data.LastName.ToLower()
        && item.Chair.ChairName.ToLower() == data.Chair.ToLower()
        && item.Chair.Faculty.FacultyName.ToLower() ==
data.Faculty.ToLower()).First()
        , unitOfWork, new HIndex(), new I10Index()).GoogleHIndex;
    }
    else
    {
        data.GoogleHIndex = new Entities.Researcher(researchers.First(),
unitOfWork, new HIndex(), new I10Index()).GoogleHIndex;
    }

    return data;
}

public static void SaveData(Data data)
{
    var localFaculty = new DatabaseLibrary.DatabaseModel.Faculty
    {
        FacultyName = data.Faculty
    };
    var localResearcher = new
DatabaseLibrary.DatabaseModel.Researcher();
    lock (unitOfWork)
    {
        var faculties = unitOfWork.Faculties.Find(item =>

```

```

        item.FacultyName.ToLower() ==
localFaculty.FacultyName.ToLower());

        if (faculties.Count() == 0)
        {
            unitOfWork.Faculties.Create(localFaculty);
            unitOfWork.Save();
        }

        else
        {
            localFaculty = faculties.First();
        }

        var localChair = new DatabaseLibrary.DatabaseModel.Chair
        {
            ChairName = data.Chair,
            FacultyId = localFaculty.FacultyId
        };

        var chairs = unitOfWork.Chairs.Find(item =>
            item.ChairName.ToLower() == localChair.ChairName.ToLower()
            && item.FacultyId == localChair.FacultyId);

        if (chairs.Count() == 0)
        {
            unitOfWork.Chairs.Create(localChair);
            unitOfWork.Save();
        }

```

```

else
{
    localChair = chairs.First();
}

localResearcher = new DatabaseLibrary.DatabaseModel.Researcher
{
    FirstName = data.FirstName,
    MiddleName = data.MiddleName,
    LastName = data.LastName,
    ChairId = localChair.ChairId
};

var researchers = unitOfWork.Researchers.Find(item =>
    item.FirstName == localResearcher.FirstName
    && item.MiddleName == localResearcher.MiddleName
    && item.LastName == localResearcher.LastName
    && item.ChairId == localResearcher.ChairId);

if (researchers.Count() == 0)
{
    unitOfWork.Researchers.Create(localResearcher);
    unitOfWork.Save();
}

else
{
    localResearcher = researchers.First();

```

```

    }
}
var orcidAccount = OrcidParser.GetResearcher(data.Orcid);

lock (unitOfWork)
{
    var account = SaveOrcidAccount(orcidAccount);

    var researcherOrcids = unitOfWork.ResearcherOrcids.Find(item =>
        item.OrcidAccountId == account.OrcidAccountId
        && item.ResearcherId == localResearcher.ResearcherId);

    if (researcherOrcids.Count() == 0)
    {
        unitOfWork.ResearcherOrcids.Create(new ResearcherOrcid
        {
            OrcidAccountId = account.OrcidAccountId,
            ResearcherId = localResearcher.ResearcherId
        });
        unitOfWork.Save();
    }
}

string userSearch = orcidAccount.FullName;
userSearch = userSearch.Replace(' ', '+').Replace(".", "");
IWebDriver driver = WebSocket.Driver;
IWebElement googleUserId;
lock (driver)
{

```

```

        driver.Url =
"https://scholar.google.com.ua/citations?hl=uk&view_op=search_authors&mauthors
="
        + userSearch + "+National+Technical+University+of+Ukraine";

        googleUserId = driver.FindElement(By.CssSelector("#gsc_sa_ccl >
div > div > a"));
    }

    var googleAccount = Task.Run(() =>
GoogleScholarParser.GetResearcher(googleUserId.GetAttribute("href").Replace("@ "
https://scholar.google.com.ua/citations?hl=uk&user=", "")));
    var publonsAccount = Task.Run(() =>
PublonsParser.GetResearcher(orcIdAccount.PublonsId));
    var scopusAccount = Task.Run(() =>
ScopusParser.GetResearcher(orcIdAccount.ScopusAuthorId));

    lock (unitOfWork)
    {
        int googleId =
ResearcherBusiness.SaveResearcher(googleAccount.Result);
        int publonsId =
ResearcherBusiness.SaveResearcher(publonsAccount.Result);
        int scopusId =
ResearcherBusiness.SaveResearcher(scopusAccount.Result);

        var researcherGoogles = unitOfWork.ResearcherAccounts.Find(item
=>

        item.AccountId == googleId

```

```

        && item.ResearcherId == localResearcher.ResearcherId);

if (researcherGoogles.Count() == 0)
{
    unitOfWork.ResearcherAccounts.Create(new ResearcherAccount
    {
        AccountId = googleId,
        ResearcherId = localResearcher.ResearcherId
    });
    unitOfWork.Save();
}

var researcherPublons = unitOfWork.ResearcherAccounts.Find(item
=>
    item.AccountId == publonsId
    && item.ResearcherId == localResearcher.ResearcherId);

if (researcherPublons.Count() == 0)
{
    unitOfWork.ResearcherAccounts.Create(new ResearcherAccount
    {
        AccountId = publonsId,
        ResearcherId = localResearcher.ResearcherId
    });
    unitOfWork.Save();
}

var researcherScopuses =
unitOfWork.ResearcherAccounts.Find(item =>

```



```

        item.AccountId == scopusId
        && item.ResearcherId == localResearcher.ResearcherId);

    if (researcherScopuses.Count() == 0)
    {
        unitOfWork.ResearcherAccounts.Create(new ResearcherAccount
        {
            AccountId = scopusId,
            ResearcherId = localResearcher.ResearcherId
        });
        unitOfWork.Save();
    }

    unitOfWork.Save();
}

public static DatabaseLibrary.DatabaseModel.OrcidAccount
SaveOrcidAccount(OrcidAccount account)
{
    DatabaseLibrary.DatabaseModel.OrcidAccount orcidAccount = new
    DatabaseLibrary.DatabaseModel.OrcidAccount
    {
        FullName = account.FullName,
        OrcId = account.OrcId,
        PublicationsCount = account.PublicationsCount,
        PublonsId = account.PublonsId,
        PublonsLink = account.PublonsLink,
        ScopusAuthorId = account.ScopusAuthorId,
    }
}

```

```
ScopusLink = account.ScopusLink
};
```

```
orcidAccount.OrcidPublications =
account.OrcidPublications.Select(item => ToOrcidPublication(item)).ToList();
```

```
var list = unitOfWork.OrcidAccounts.Find(item => item.OrcId ==
orcidAccount.OrcId).ToList();
```

```
if (list.Count == 0)
```

```
{
    unitOfWork.OrcidAccounts.Create(orcidAccount);
    unitOfWork.Save();
}
```

```
else
```

```
{
    orcidAccount.OrcidAccountId =
unitOfWork.OrcidAccounts.Find(item => item.OrcId ==
orcidAccount.OrcId).First().OrcidAccountId;
}
```

```
var publicationsList =
unitOfWork.OrcidPublications.GetAll().Where(item => item.OrcidAccountId ==
orcidAccount.OrcidAccountId).ToList();
```

```
foreach (var publication in orcidAccount.OrcidPublications)
{
```

```
    var accountPublication = publicationsList.Find(item => item.Title ==
    publication.Title);
```

```
    if (accountPublication == null)
```

```
        {
            unitOfWork.OrcidPublications.Create(publication);
            unitOfWork.Save();
        }
    }

    return orcidAccount;
}

public static DatabaseLibrary.DatabaseModel.OrcidPublication
ToOrcidPublication(OrcidPublication publication)
{
    return new DatabaseLibrary.DatabaseModel.OrcidPublication
    {
        Date = publication.Date,
        Title = publication.Title
    };
}
}
```

**Факультет інформатики та обчислювальної техніки**  
**Кафедра автоматизованих систем обробки інформації і управління**

**“ЗАТВЕРДЖЕНО”**

**В.о. завідувача кафедри**

\_\_\_\_\_ **О.А. Павлов**

**“    ”    \_\_\_\_\_ 2019 р.**

**Програмне забезпечення оцінювання результатів наукової**  
**діяльності викладачів**

**Керівництво користувача**

**КП.ІП- 5104.045490.05.34**

**“ПОГОДЖЕНО”**

**Керівник проекту:**

\_\_\_\_\_ **О.Д. Фіногенов**

**Нормоконтроль:**

\_\_\_\_\_ **Ліщук К.І.**

**Виконавець:**

\_\_\_\_\_ **В.Ю. Гладишко**

**Київ – 2019 року**

## ЗМІСТ

<b>1</b>	<b>КЕРІВНИЦТВО КОРИСТУВАЧА .....</b>	<b>3</b>
1.1	СТВОРЕННЯ, РЕДАГУВАННЯ ТА ВИДАЛЕННЯ ДАНИХ ПРО ФАКУЛЬТЕТИ .....	3
1.2	СТВОРЕННЯ, РЕДАГУВАННЯ ТА ВИДАЛЕННЯ ДАНИХ ПРО КАФЕДРИ .....	7
1.3	СТВОРЕННЯ, РЕДАГУВАННЯ ТА ВИДАЛЕННЯ ДАНИХ ПРО ВИКЛАДАЧІВ.....	11
1.4	ПЕРЕВІРКА ДАНИХ З ФАЙЛУ .....	16
1.5	ЗАВАНТАЖЕННЯ ДАНИХ З ФАЙЛУ .....	18
1.6	ПЕРЕГЛЯД СТАТИСТИКИ ПО ФАКУЛЬТЕТАМ .....	20
1.7	ПЕРЕГЛЯД СТАТИСТИКИ ПО КАФЕДРАМ .....	22
1.1	ПЕРЕГЛЯД СТАТИСТИКИ ПО ВИКЛАДАЧАМ.....	23

# 1 КЕРІВНИЦТВО КОРИСТУВАЧА

Користувач має наступні можливості в настільному додатку:

- створення, редагування та видалення даних про факультети;
- створення, редагування та видалення даних про кафедри;
- створення, редагування та видалення даних про викладачів;
- перевірка даних з файлу;
- завантаження даних з файлу;
- перегляд статистики по факультетам;
- перегляд статистики по кафедрам;
- перегляд статистики по викладачам.

## 1.1 Створення, редагування та видалення даних про факультети

Спочатку потрібно відкрити запусити настільний додаток, рисунок 1.1.

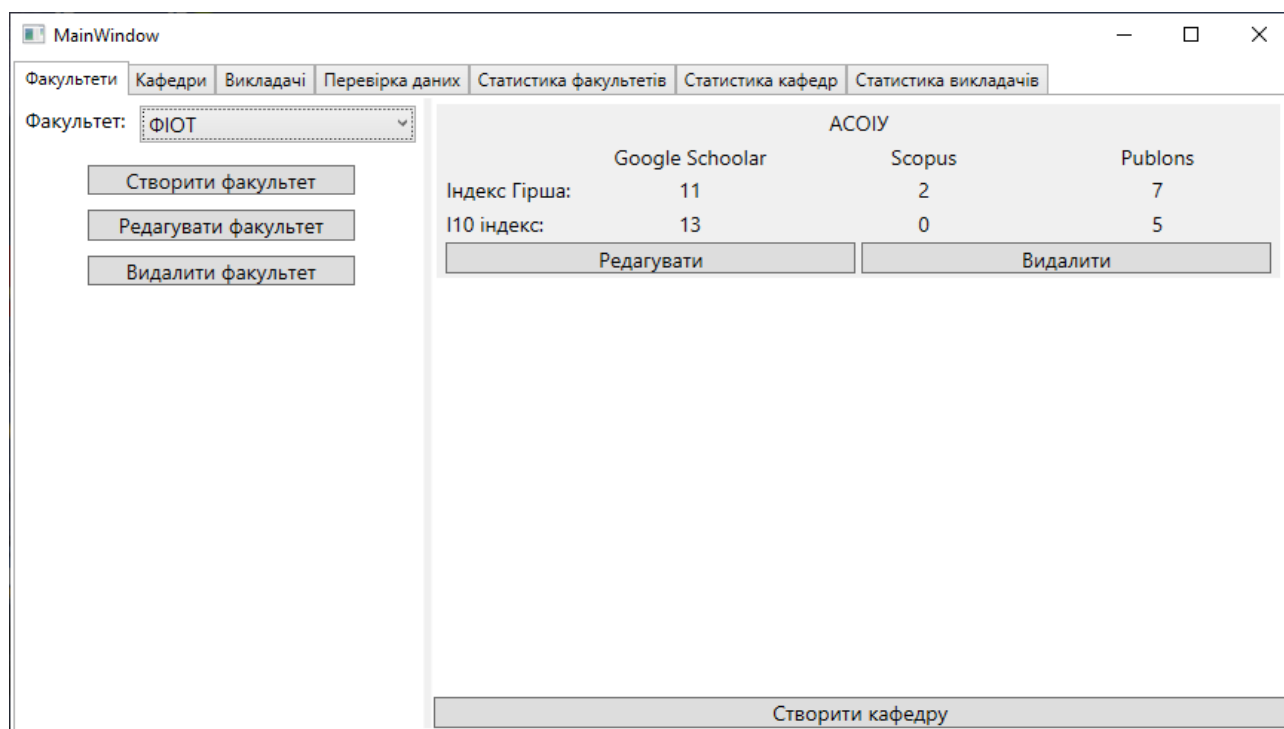


Рисунок 1.1 – Настільний додаток

Для створення даних про факультет, користувач повинен натиснути на кнопку «Створити факультет». Після натиснення кнопки відкривається форма створення факультету – рисунок 1.2.

Рисунок 1.2 – Форма створення факультету

Далі користувач повинен ввести назву факультету та натиснути кнопку «Створити» – Рисунок 1.3.

Рисунок 1.3 – Створення факультету

Після створення факультету користувач опиняється на головному вікні додатку – Рисунок 1.1.

Для редагування даних про факультет, користувач повинен вибрати факультет з випадаючого списку – Рисунок 1.4.

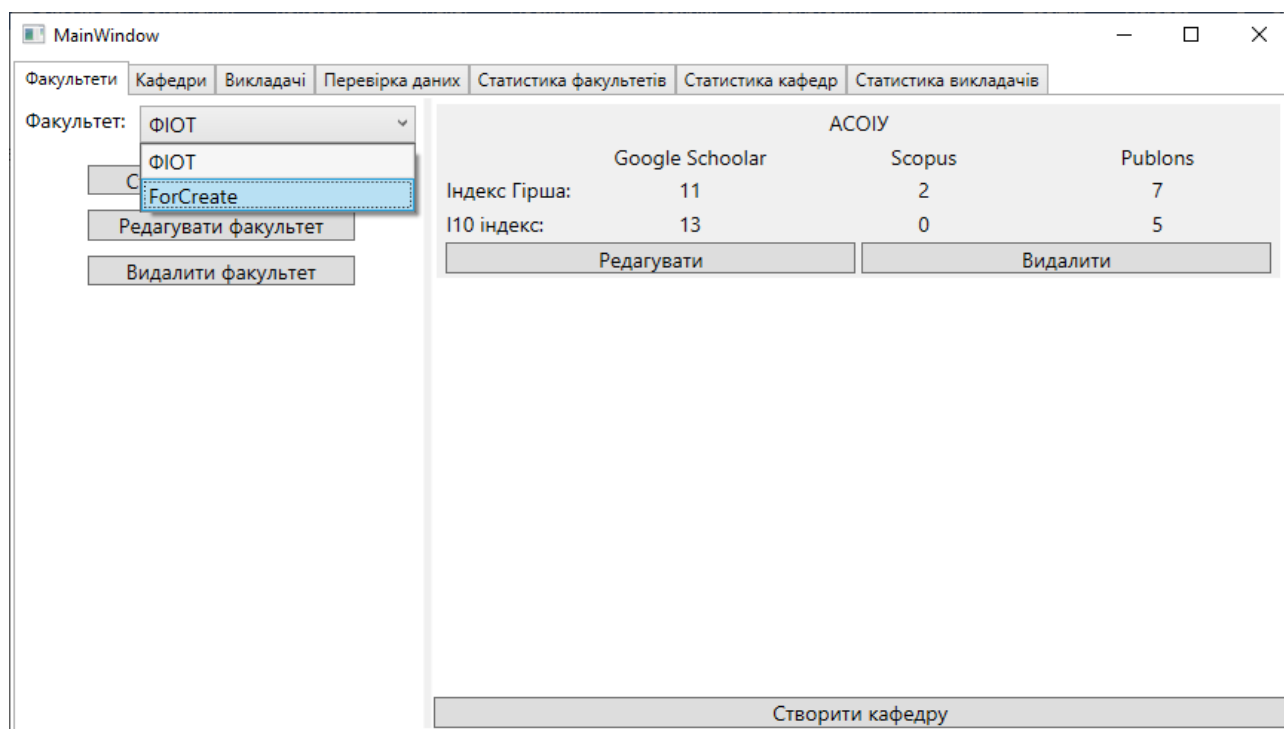


Рисунок 1.4 – Вибір факультету з випадального списку

Далі необхідно натиснути кнопку «Редагувати факультет», після цього відкриється форма редагування факультету – Рисунок 1.5.

Рисунок 1.5 – Форма редагування факультету

Потім користувач може відредагувати назву факультету, і після цього треба натиснути кнопку «Редагувати» – Рисунок 1.6.



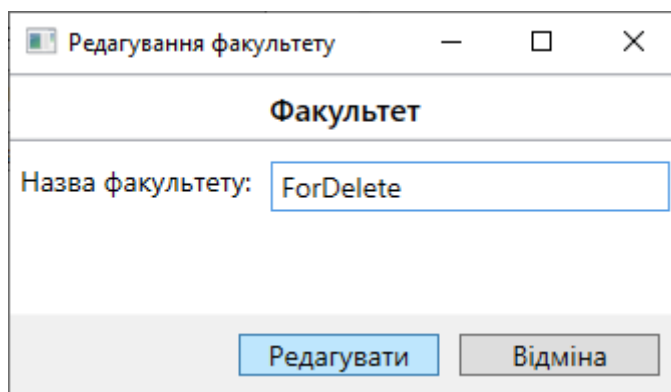


Рисунок 1.6 – Редагування факультету

Після редагування факультету користувач опиняється на головному вікні додатку – Рисунок 1.1.

Для видалення даних про факультет потрібно вибрати факультет з випадаючого списку – Рисунок 1.4, а потім необхідно натиснути кнопку «Видалити факультет», після цього з'явиться вікно для підтвердження – Рисунок 1.7.

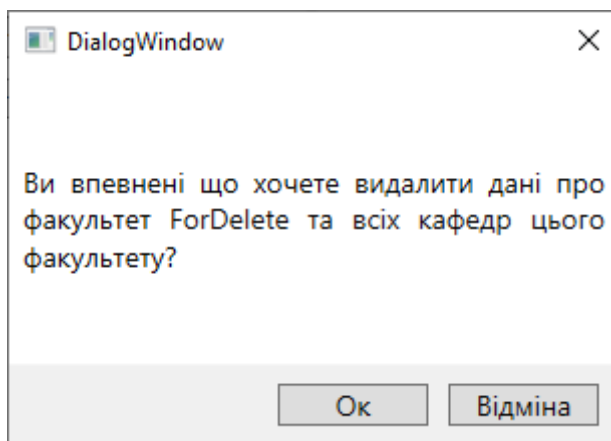


Рисунок 1.7 – Вікно підтвердження видалення факультету

Далі користувач повинен натиснути кнопку «Ок», після цього запис видалиться і користувач опиниться на головному вікні – Рисунок 1.1.

## 1.2 Створення, редагування та видалення даних про кафедри

Для створення кафедри необхідно перейти на вкладку «Кафедри» – Рисунок 1.8.

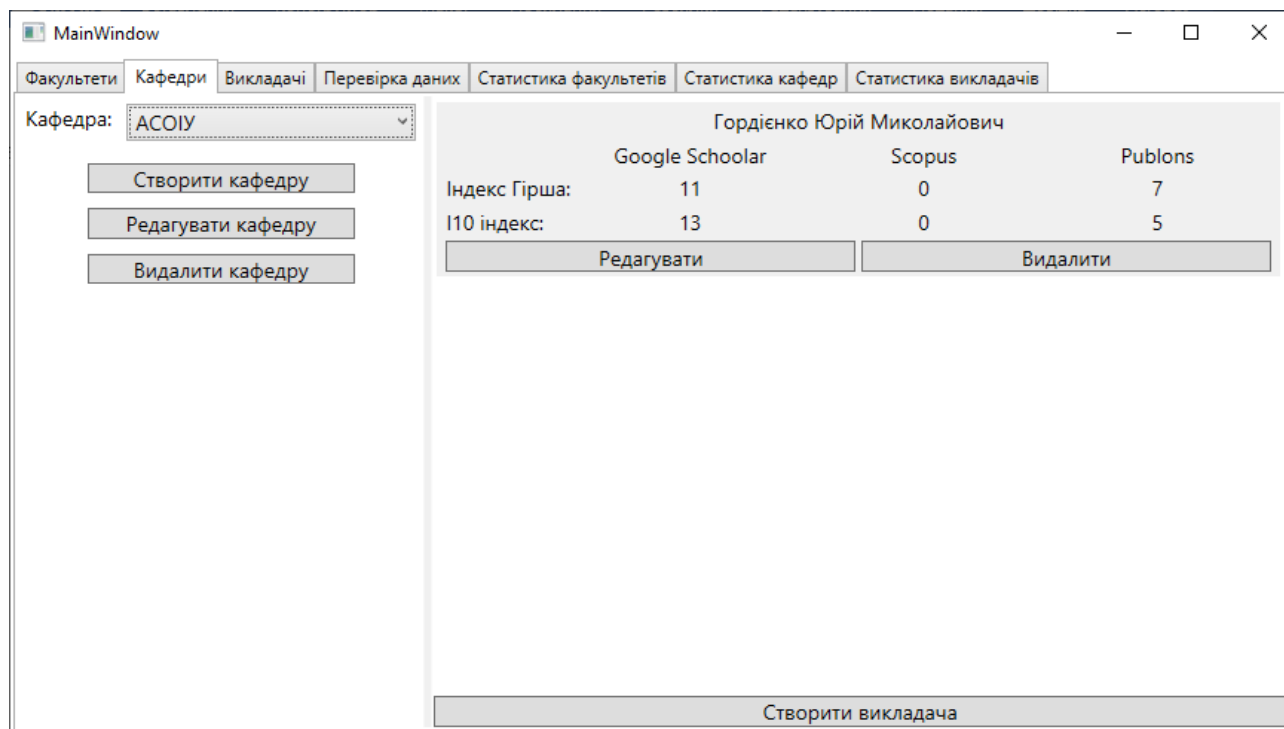


Рисунок 1.8 – Вкладка «Кафедри»

Далі необхідно натиснути кнопку «Створити кафедру», після цього відкриється форма створення кафедри – Рисунок 1.9.

Рисунок 1.9 – Форма створення кафедри

Далі користувач повинен вибрати факультет з випадаючого списку і ввести назву кафедри, після цього необхідно натиснути кнопку «Створити» – Рисунок 1.10.

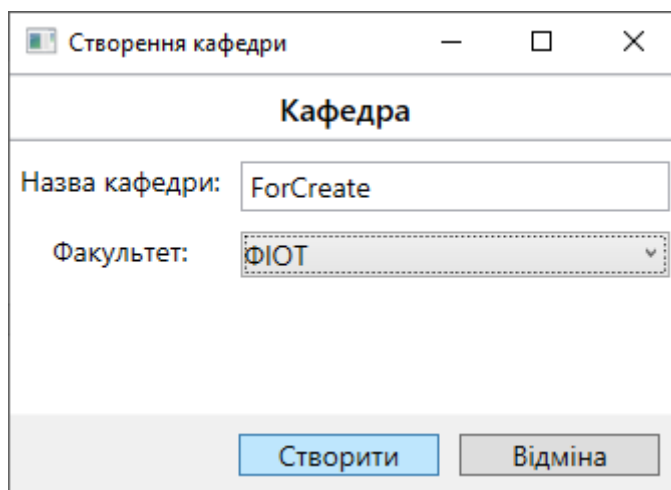


Рисунок 1.10 – Створення кафедри

Після редагування факультету користувач опиняється на вкладці «Кафедри» – Рисунок 1.8.

Для редагування кафедри, користувач повинен вибрати кафедру з випадаючого списку – Рисунок 1.11.

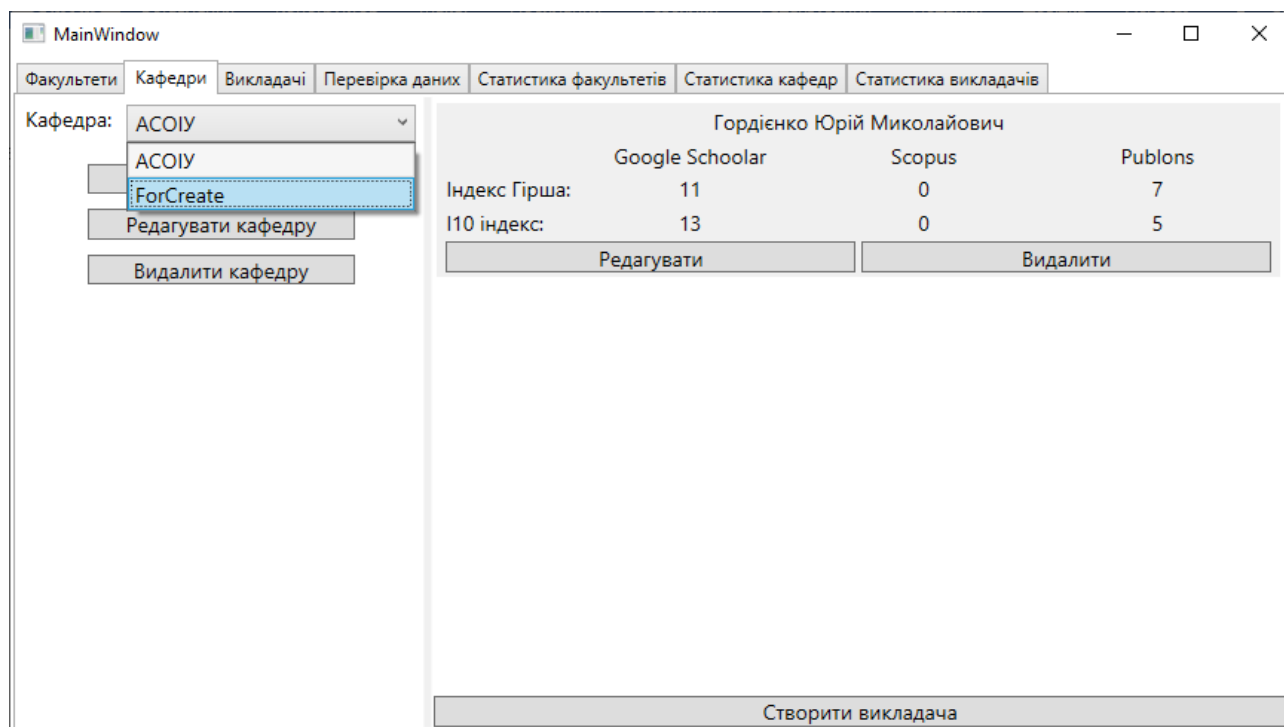


Рисунок 1.11 – Вибір кафедри з випадаючого списку

Далі необхідно натиснути кнопку «Редагувати кафедру», після цього відкриється форма редагування кафедри – Рисунок 1.12.

Рисунок 1.12 – Форма редагування кафедри

Далі користувач може змінити факультет, до якого належить кафедра, або змінити назву кафедри. Після внесених змін необхідно натиснути кнопку «Редагувати кафедру» – Рисунок 1.13.

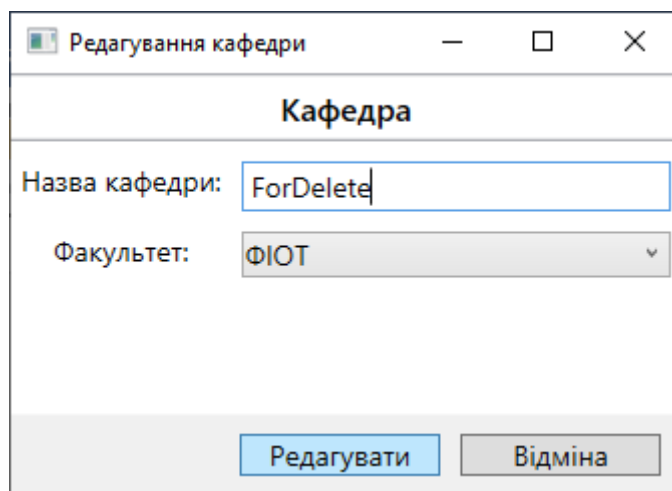


Рисунок 1.13 – Редагування кафедри

Після редагування кафедри користувач опиняється на вкладці «Кафедри» – Рисунок 1.8.

Для видалення кафедри користувач повинен вибрати кафедру з випадаючого списку – Рисунок 1.11, а далі натиснути кнопку «Видалити кафедру», після цього з'явиться вікно для підтвердження видалення кафедри – Рисунок 1.14.

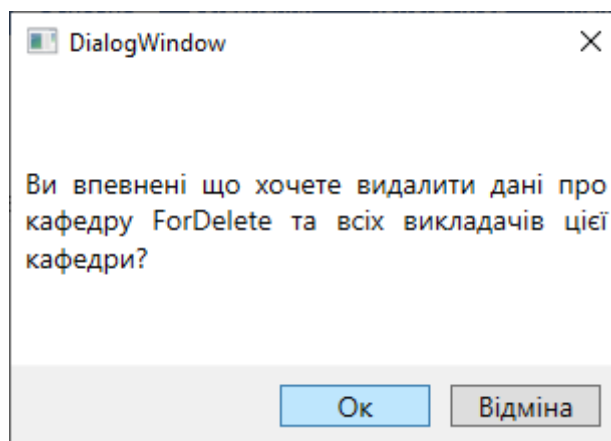


Рисунок 1.14 – Вікно підтвердження видалення кафедри

Далі користувач повинен підтвердити видалення натиснувши кнопку «Ок», після цього користувач опиняється на вкладці «Кафедри» – Рисунок 1.8.

### 1.3 Створення, редагування та видалення даних про викладачів

Для створення викладача користувачу необхідно перейти на вкладку «Викладачі» – Рисунок 1.15.

Рисунок 1.15 – Вкладка «Викладачі»

Далі необхідно натиснути кнопку «Створити викладача», після цього відкриється форма створення викладача – Рисунок 1.16.

Рисунок 1.16 – Форма створення викладача

Далі користувач вводить прізвище, ім'я та ім'я по-батькові у відповідні поля, після цього вибирає кафедру з випадаючого списку і нажимає кнопку «Створити» – Рисунок 1.17.

Рисунок 1.17 – Створення викладача

Після створення викладача користувач повертається на вкладку «Викладачі» – Рисунок 1.15.

Для того, щоб редагувати викладача користувачу необхідно здійснити пошук викладача – Рисунок 1.18, або перейти на вкладку «Кафедри» – Рисунок 1.8.

Муха Ірина Павлівна			
	Google Scholar	Scopus	Publons
Індекс Гірша:	0	0	0
I10 індекс:	0	0	0
Редагувати		Видалити	

Створити викладача

Рисунок 1.18 – Пошук викладача

Далі користувач повинен вибрати викладача і натиснути кнопку «Редагувати» – Рисунок 1.19.



MainWindow

Факультети Кафедри Викладачі **Перевірка даних** Статистика факультетів Статистика кафедр Статистика викладачів

Прізвище: М

Ім'я:

По-Батькові:

Пошук

Муха Ірина Павлівна

	Google Scholar	Scopus	Publons
Індекс Гірша:	0	0	0
I10 індекс:	0	0	0

Редагувати Видалити

Створити викладача

Рисунок 1.19 – Вибір викладача в результатах пошуку

Після натиснення кнопки відкривається форма редагування викладача – Рисунок 1.20.

Редагування викладача

**Викладач**

Прізвище: Муха

Ім'я: Ірина

По-батькові: Павлівна

Кафедра: АСОІУ

Редагувати Відміна

Рисунок 1.20 – Форма редагування викладача

Далі користувач може змінити прізвище, ім'я, ім'я по-батькові, кафедру, до якої належить викладач. Після внесених змін необхідно натиснути кнопку «Редагувати», після цього користувач опиняється на вкладці «Викладачі» – Рисунок 1.18.

Для видалення викладача користувач повинен спочатку вибрати викладача за допомогою пошуку – Рисунок 1.18, або вибрати викладача в вкладці «Кафедри», далі необхідно натиснути кнопку «Видалити» – Рисунок 1.21.

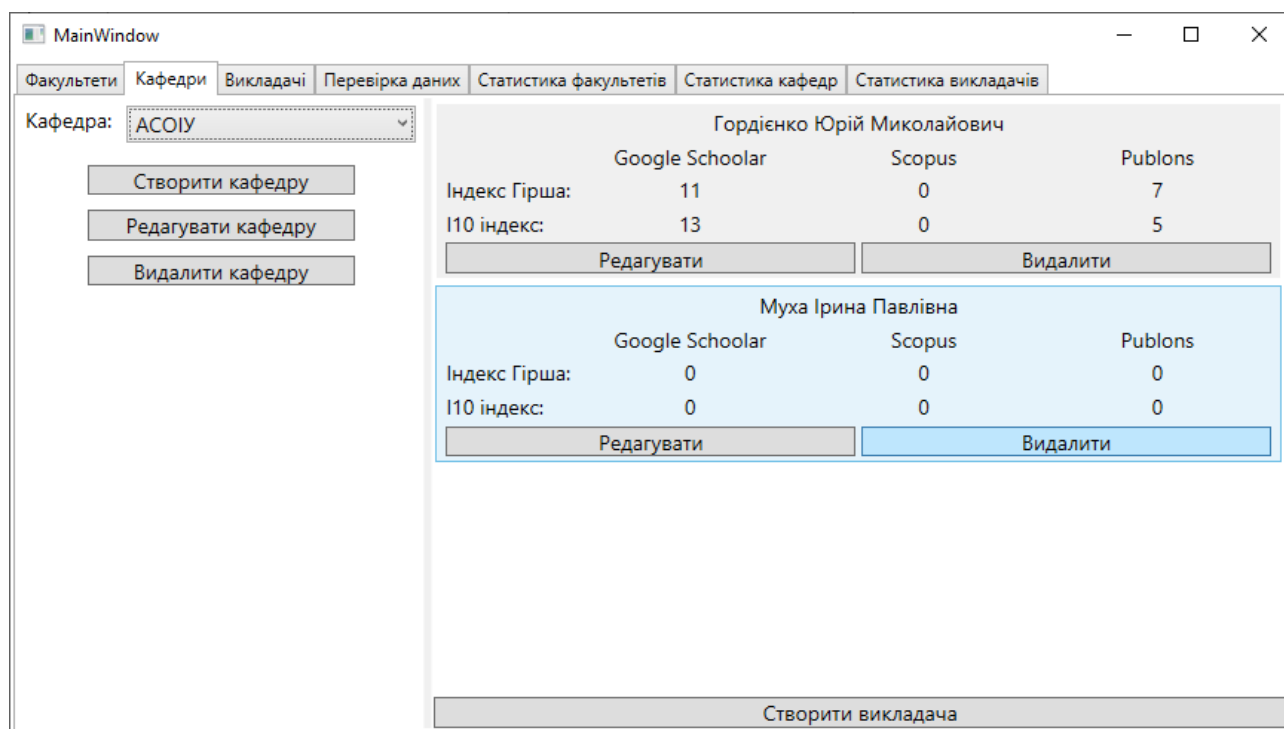


Рисунок 1.21 – Вибір викладача в кафедрі

Після натиснення кнопки з'явиться вікно для підтвердження видалення викладача – Рисунок 1.22.

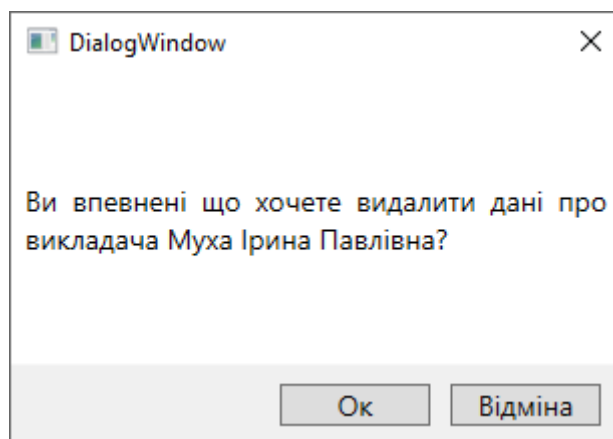


Рисунок 1.22 – Вікно підтвердження видалення викладача

Далі користувач повинен підтвердити видалення натиснувши кнопку «Ок», після цього користувач опиняється на вкладці «Викладачі» – Рисунок 1.18.

#### 1.4 Перевірка даних з файлу

Для перевірки даних користувач спочатку повинен перейти на вкладку «Перевірка даних» та натиснути кнопку «Завантажити дані для перевірки» – Рисунок 1.23.

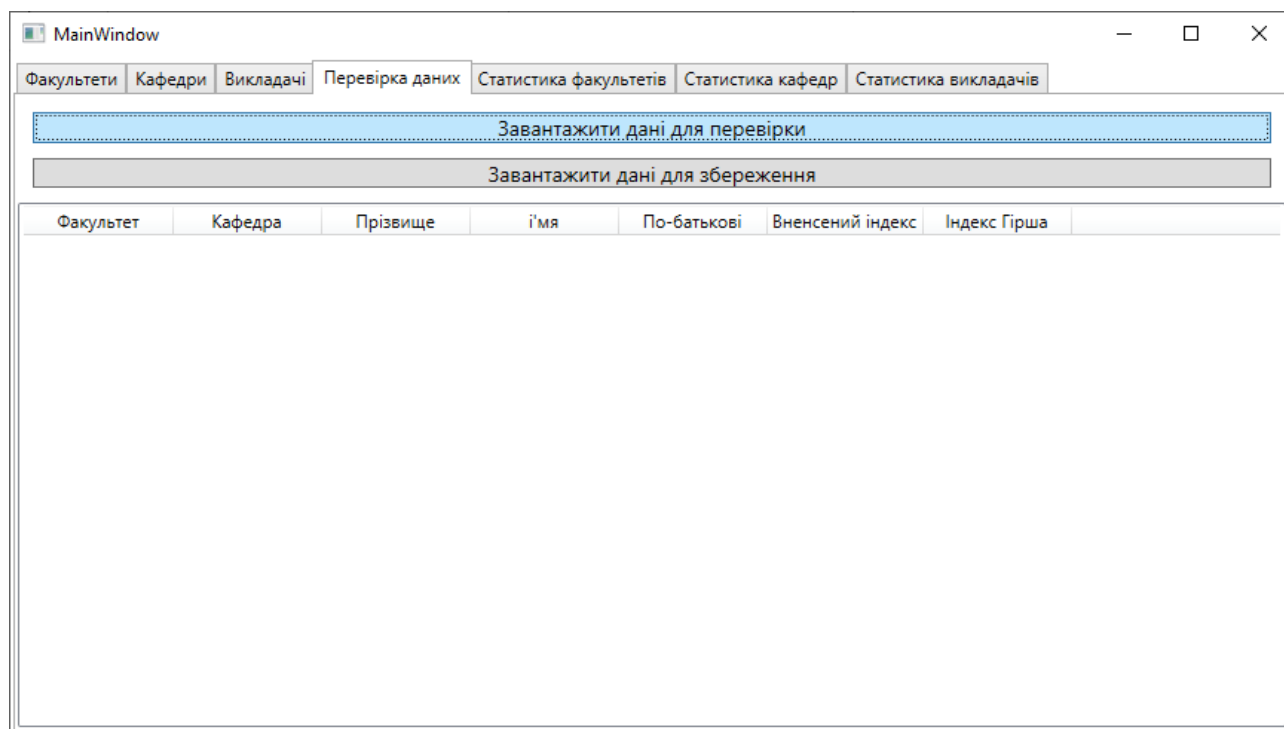


Рисунок 1.23 – Вкладка «Перевірка даних»

Після натиснення кнопки відкриється вікно для вибору файлів, користувач має обрати файл для перевірки – Рисунок 1.24.

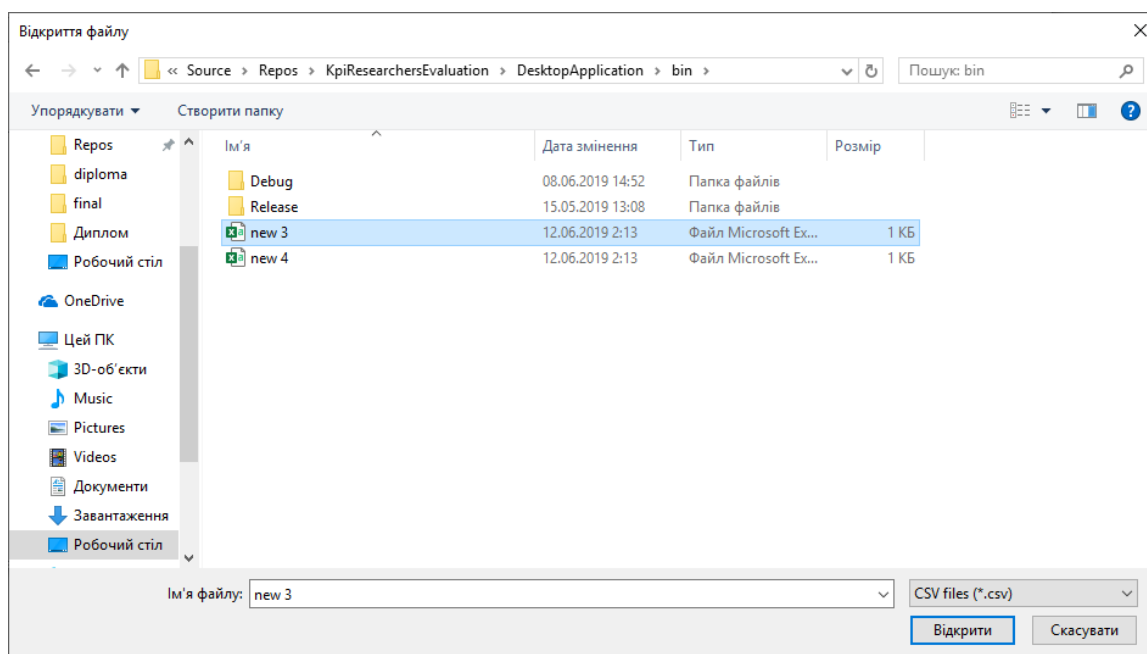


Рисунок 1.24 – Вибір файлу для перевірки

Далі користувач повинен вибрати файл і натиснути кнопку «Відкрити», після обробки даних користувач побачить результат – Рисунок 1.25.

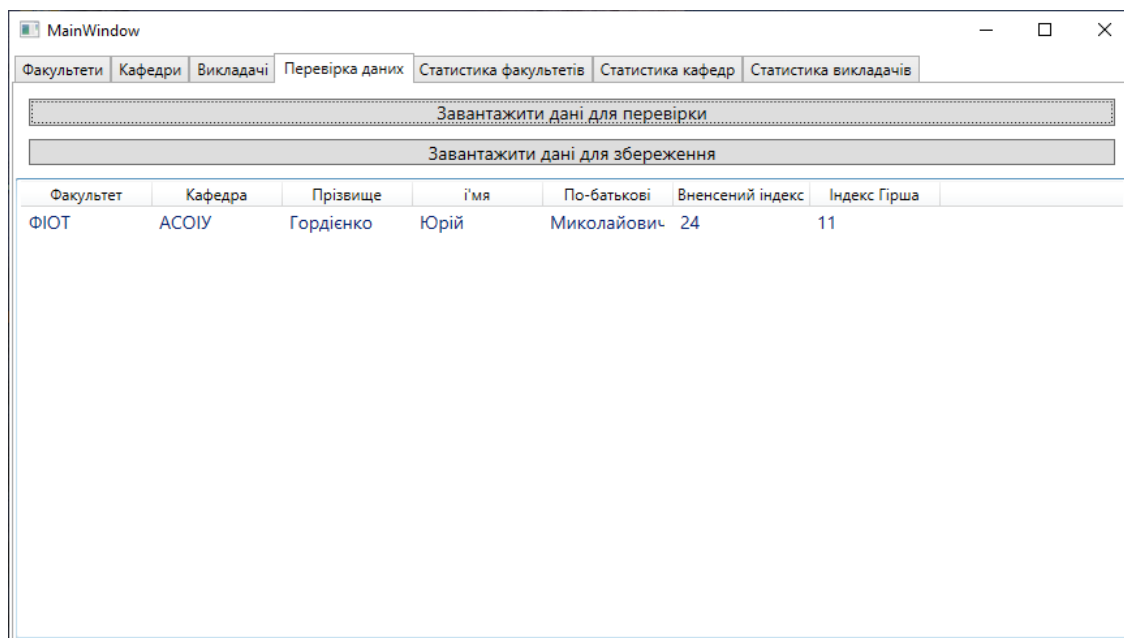


Рисунок 1.25 – Результат перевірки

### 1.5 Завантаження даних з файлу

Для перевірки даних користувач спочатку повинен перейти на вкладку «Перевірка даних» та натиснути кнопку «Завантажити дані для збереження» – Рисунок 1.26.

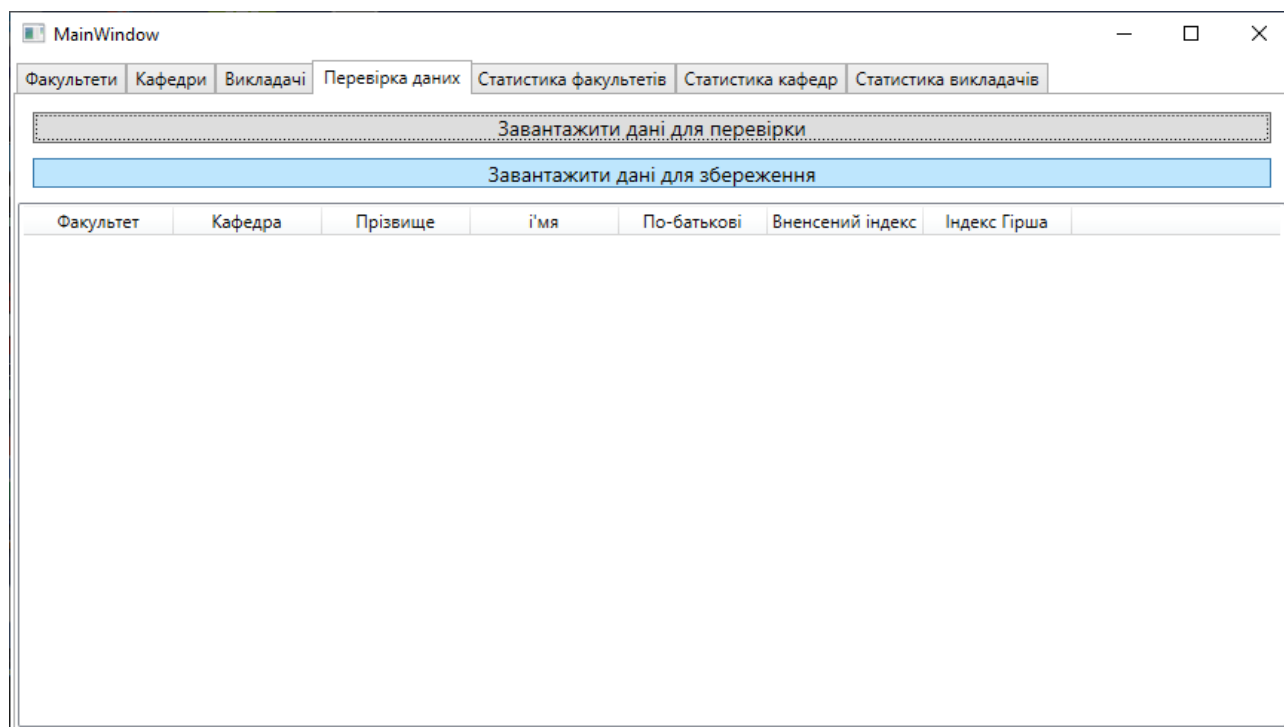


Рисунок 1.26 – Вибір файлу для завантаження

Після натиснення кнопки відкріється вікно для вибору файлів, користувач має обрати файл для завантаження – Рисунок 1.27.

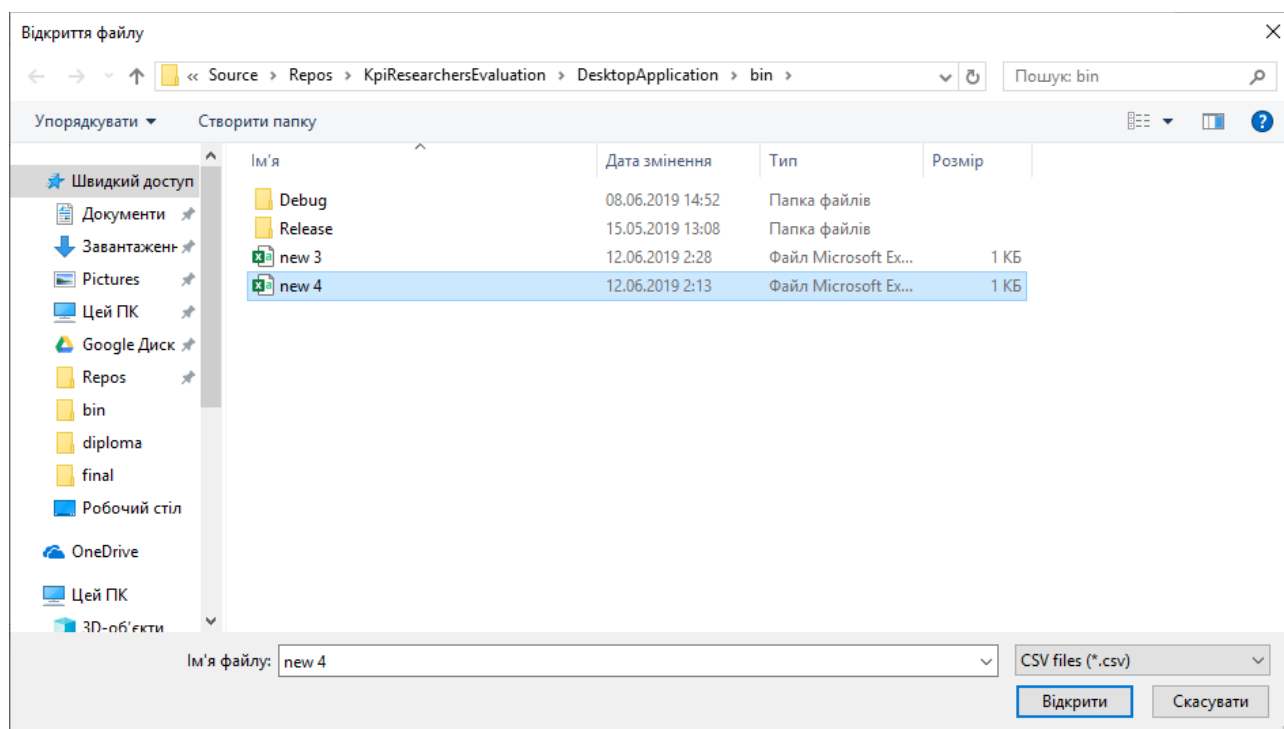


Рисунок 1.27 – Вибір файлу для завантаження

Далі користувач повинен вибрати файл і натиснути кнопку «Відкрити», після завантаження даних користувач побачить результат – Рисунок 1.28.

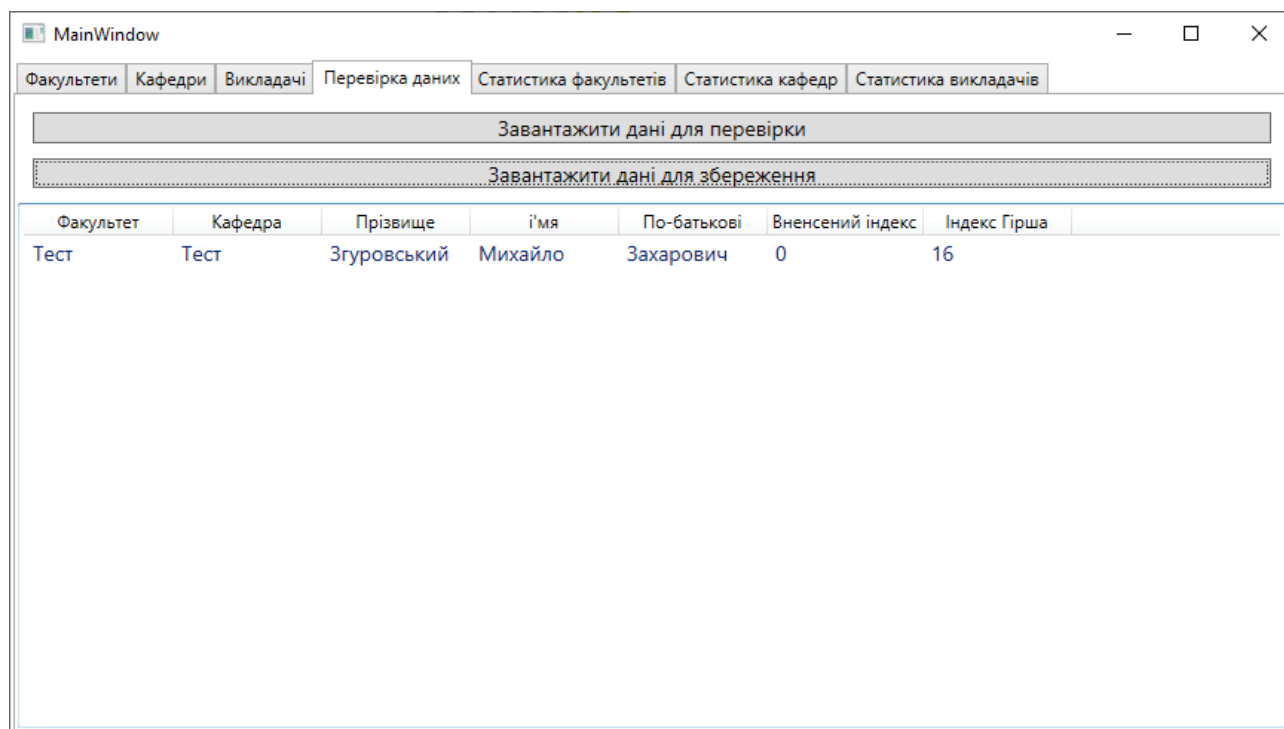


Рисунок 1.28 – Результати завантаження

## 1.6 Перегляд статистики по факультетам

Для перегляду статистики по факультетам користувачу потрібно перейти на вкладку «Статистика факультетів» – Рисунок 1.29.

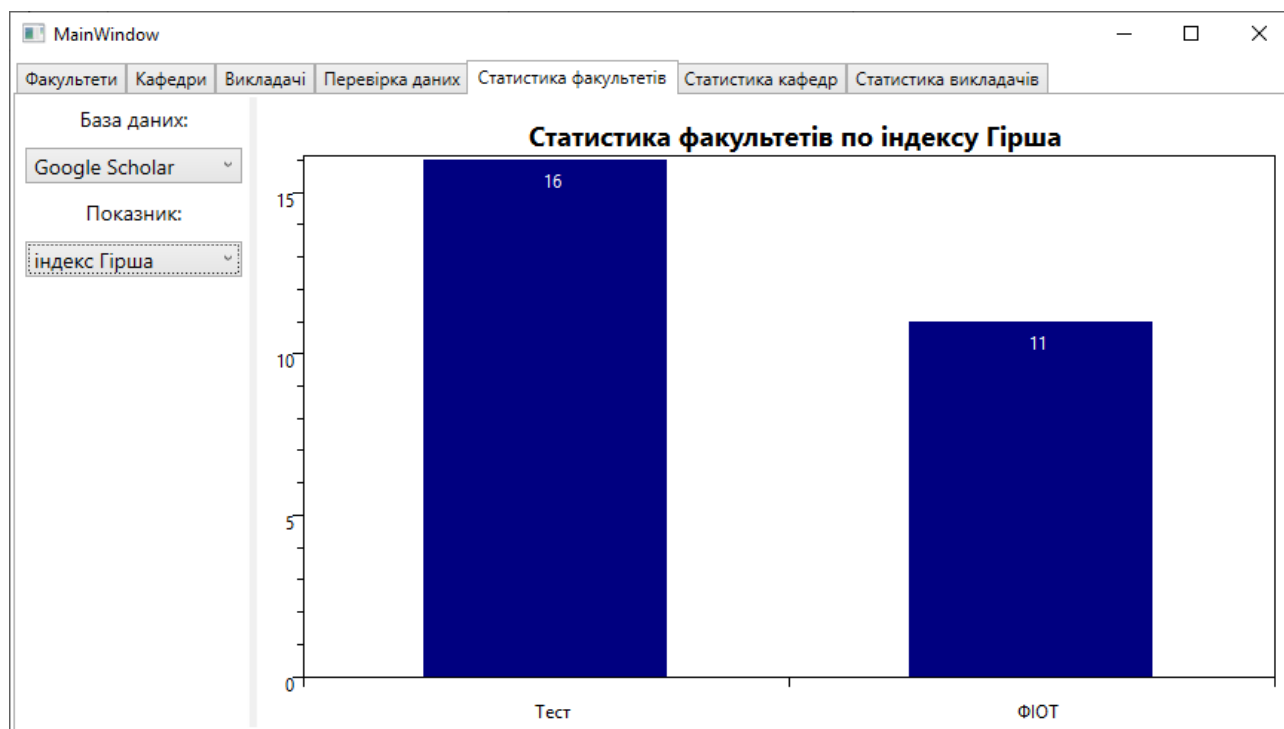


Рисунок 1.29 – Вкладка «Статистика факультетів»

Користувач може змінити або базу даних, з якої беруться показники, або показник, за яким порівнюється, за допомогою випадających списків – Рисунок 1.30.

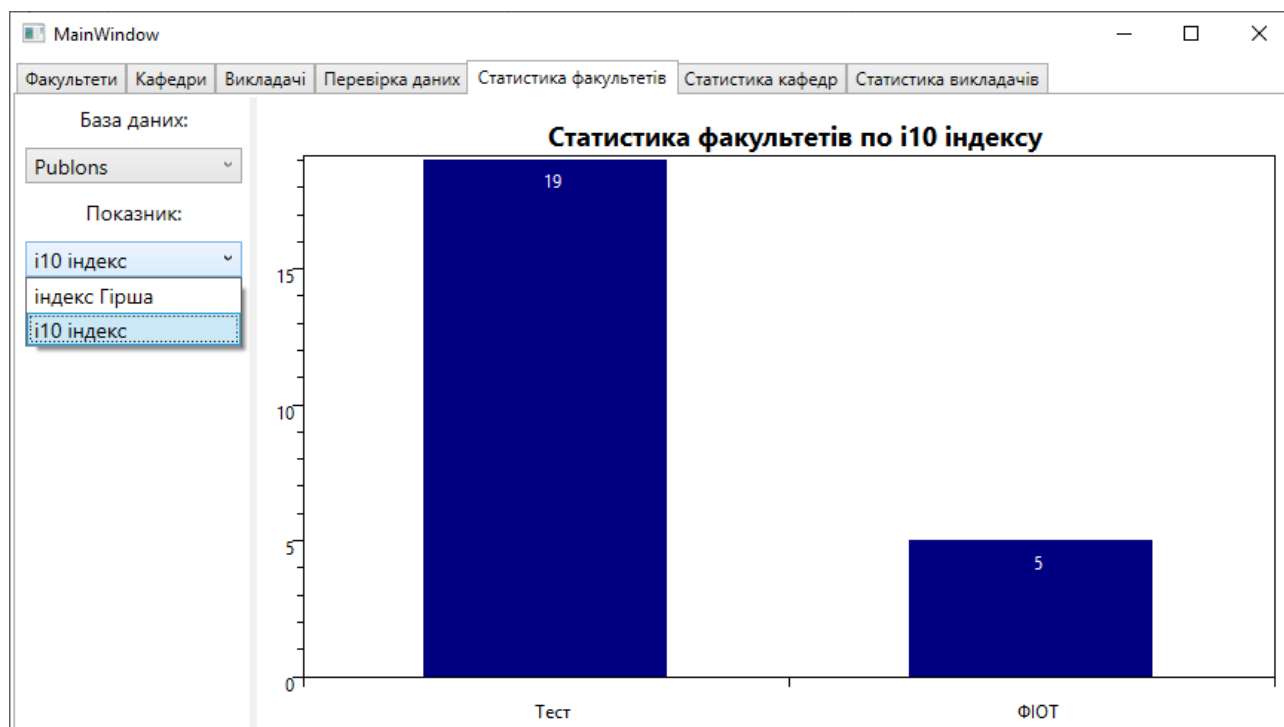




Рисунок 1.30 – Статистика фвкультетів по показнику «i10 індекс»

## 1.7 Перегляд статистики по кафедрам

Для перегляду статистики по кафедрам користувачу потрібно перейти на вкладку «Статистика кафедр» – Рисунок 1.31.



Рисунок 1.31 – Вкладка «Статистика кафедр»

Користувач може змінити або факультет, кафедри якого порівнюються, або базу даних, з якої беруться показники, або показник, за яким порівнюється, за допомогою випадючих списків – Рисунок 1.32.



Рисунок 1.32 – Статистика кафедр факультету «Тест» по показнику «i10 індекс»

## 1.8 Перегляд статистики по викладачам

Для перегляду статистики по викладачам користувачу потрібно перейти на вкладку «Статистика викладачів» – Рисунок 1.33.



Рисунок 1.33 – Вкладка «Статистика викладачів»

Користувач може змінити або кафедру, викладачі якої порівнюються, або базу даних, з якої беруться показники, або показник, за яким порівнюється, за допомогою випадаючих списків – Рисунок 1.34.

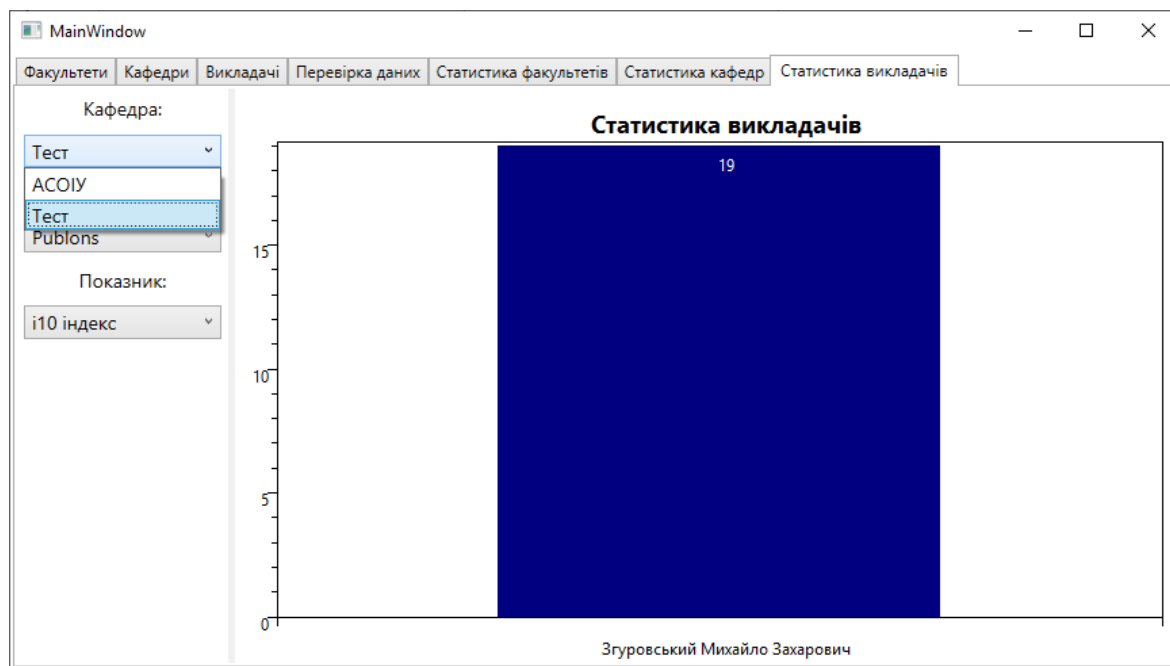


Рисунок 1.34 – Статистика викладачів кафедри «Тест» по показнику «i10 індекс»

**Факультет інформатики та обчислювальної техніки**  
**Кафедра автоматизованих систем обробки інформації і управління**

**“ЗАТВЕРДЖЕНО”**

В.о. завідувача кафедри

\_\_\_\_\_ О.А. Павлов

“ \_\_\_\_ ” \_\_\_\_\_ 2019 р.

**Програмне забезпечення оцінювання результатів наукової**

**діяльності викладачів**

**Графічний матеріал**

КПІ.ІІ-5104.045490.06.99

**“ПОГОДЖЕНО”**

Керівник проекту:

\_\_\_\_\_ О.Д. Фіногенов

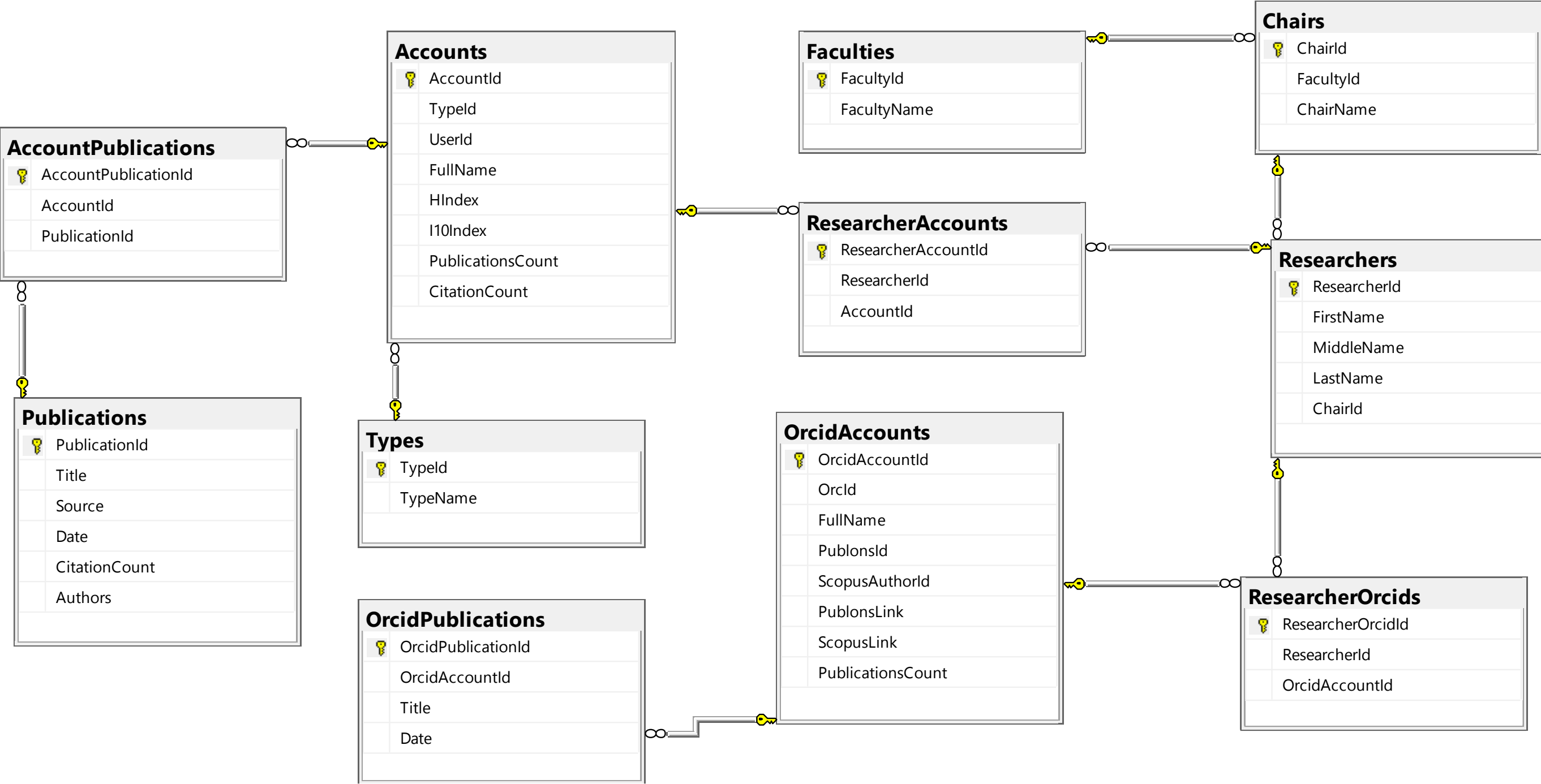
Нормоконтроль:

\_\_\_\_\_ Ліщук К.І.

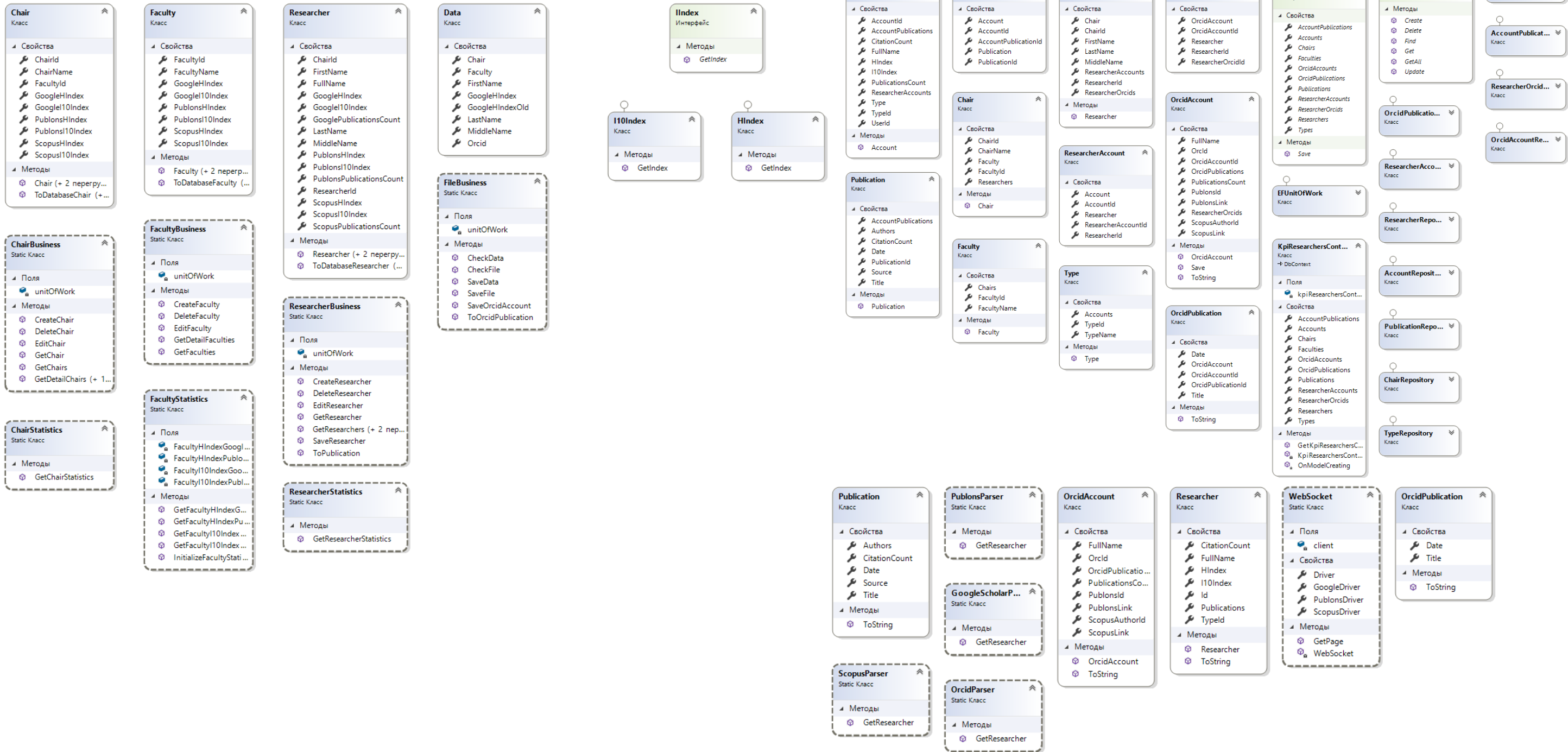
Виконавець:

\_\_\_\_\_ В.Ю. Гладишко

Київ – 2019 року



					КПІ.ІП-5104.045490.06.99.СБД					
					Схема бази даних	Літера			Маса	Масштаб
Зм.	Арк.	№ документа	Підпис	Дата						
Розробив		Гладишко В.Ю.								
Перевірів		Фіногенов О.Д.								
Т. кон.										
						Аркуш			Аркушів	
Н. кон.		Ліщук К.І.			Програмне забезпечення оцінювання результатів наукової діяльності викладачів	КПІ ім.Ігоря Сікорського Кафедра АСОІУ гр. ІП-51				
Затвердив		Фіногенов О.Д.								



					КПІ.ІП-5104.045490.06.99.СС						
					Схема структурна класів програмного забезпечення	Літера			Маса	Масштаб	
Зм.	Арк.	№ документа	Підпис	Дата							
Розробив		Гладишко В.Ю.									
Перевірив		Фіногенов О.Д.									
Т. кон.											
					Програмне забезпечення оцінювання результатів наукової діяльності викладачів	Аркуш			Аркушів		
Н. кон.		Ліщук К.І.				КПІ ім.Ігоря Сікорського Кафедра АСОІУ гр. ІП-51					
Затвердив		Фіногенов О.Д.									

MainWindow

ФаккультетиКафедриВикладачіПеревірка данихСтатистика факкультетівСтатистика кафедрСтатистика викладачів

Факкультет: ФІОТ

Створити факкультетРедагувати факкультетВидалити факкультет

ACOІУ

Google ScholarScopusPublons

Індекс Гірша: 1127І10 індекс: 1305

РедагуватиВидалити

Створити кафедру

MainWindow

ФаккультетиКафедриВикладачіПеревірка данихСтатистика факкультетівСтатистика кафедрСтатистика викладачів

Прізвище: ГордієнкоІм'я:По-Батькові:Пошук

Гордієнко Юрій Миколайович

Google ScholarScopusPublons

Індекс Гірша: 1177І10 індекс: 1305

РедагуватиВидалити

Створити викладача

MainWindow

ФаккультетиКафедриВикладачіПеревірка данихСтатистика факкультетівСтатистика кафедрСтатистика викладачів

Кафедра: ACOІУ

Створити кафедруРедагувати кафедруВидалити кафедру

Гордієнко Юрій Миколайович

Google ScholarScopusPublons

Індекс Гірша: 1107І10 індекс: 1305

РедагуватиВидалити

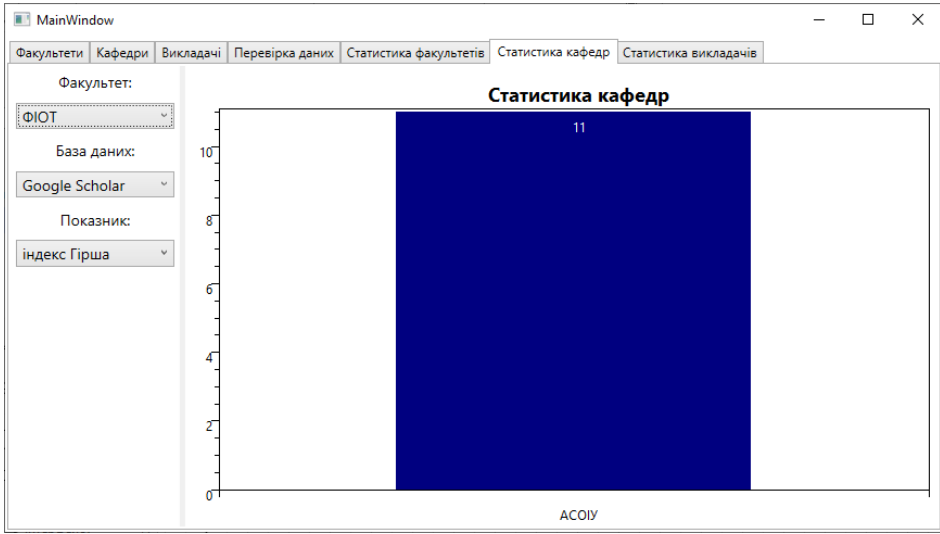
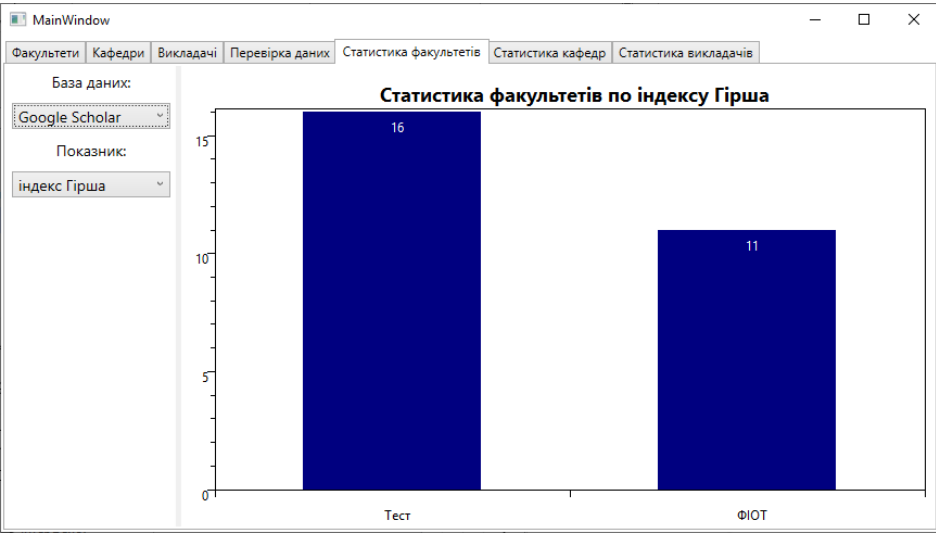
Створити викладача

MainWindow

ФаккультетиКафедриВикладачіПеревірка данихСтатистика факкультетівСтатистика кафедрСтатистика викладачів

Завантажити дані для перевіркиЗавантажити дані для збереження

Факкультет	Кафедра	Прізвище	Ім'я	По-батькові	Внесенний індекс	Індекс Гірша
ФІОТ	ACOІУ	Гордієнко	Юрій	Миколайович	24	11



Створення викладача

Викладач

Прізвище:Ім'я:По-батькові:Кафедра: ACOІУ

СтворитиВідміна

Редагування викладача

Викладач

Прізвище: ГордієнкоІм'я: ЮрійПо-батькові: МиколайовичКафедра: ACOІУ

РедагуватиВідміна

Створення кафедри

Кафедра

Назва кафедри:Факкультет: ФІОТ

СтворитиВідміна

Редагування кафедри

Кафедра

Назва кафедри: ACOІУФаккультет: ФІОТ

РедагуватиВідміна

Створення факкультету

Факкультет

Назва факкультету:

СтворитиВідміна

Редагування факкультету

Факкультет

Назва факкультету: ФІОТ

РедагуватиВідміна

DialogWindow

Ви впевнені що хочете видалити дані про факкультет ФІОТ та всіх кафедр цього факкультету?

ОкВідміна

					КПІ.ІП-5104.045490.06.99.KE				
					Креслення вигляду екранних форм				
Зм.	Арк.	№ документа	Підпис	Дата					
Розробив		Гладишко В.Ю.			Програмне забезпечення оцінювання результатів наукової діяльності викладачів				
Перевірів		Фіногенов О.Д.							
Т. кон.									
					КПІ ім.Ігоря Сікорського Кафедра ACOІУ гр. ІП-51				
Н. кон.		Ліщук К.І.							
Затвердив		Фіногенов О.Д.							